Online 2 (Advanced SQL)

Section: C

Duration: 1 hour 20 minutes

Problem 1:

Print information of all the managers (employees that manage other employees) whose managed employees do the same job as him/her. Use **join** (50% marks will be deducted if done using subquery). Sample output:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	...	MANAGER_ID	DEPARTMENT_ID
103	Alexander	Hunold	...	102	60

Solution:

```
SELECT DISTINCT M.*
FROM HR.EMPLOYEES E JOIN HR.EMPLOYEES M ON E.MANAGER_ID = M.EMPLOYEE_ID
WHERE E.JOB_ID = M.JOB_ID;
```

Problem 2:

Print information of the employees that are the only employees in their job. Order the results by employee id. Use **subquery** (50% marks will be deducted if done using join). Sample output:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	...	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	...	-	90
108	Nancy	Greenberg	...	101	100
...	...	...	...	...	...

10 rows

Solution:

```
SELECT *
FROM HR.EMPLOYEES E
WHERE NOT EXISTS (
    SELECT *
    FROM HR.EMPLOYEES
    WHERE EMPLOYEE_ID != E.EMPLOYEE_ID AND JOB_ID = E.JOB_ID
)
ORDER BY E.EMPLOYEE_ID;
```

Problem 3:

Find all the departments that have at least one employee who was hired before the manager of the department. For example, consider the following one row of sample output. It implies that the shipping department has at least one employee who was hired before the manager of the shipping department. Sort the result by department id.

DEPARTMENT_ID	DEPARTMENT_NAME
50	Shipping

Solution:

```
SELECT D.DEPARTMENT_ID, D.DEPARTMENT_NAME
FROM HR.DEPARTMENTS D JOIN HR.EMPLOYEES M ON D.MANAGER_ID = M.EMPLOYEE_ID
WHERE M.HIRE_DATE > ANY (
    SELECT HIRE_DATE
    FROM HR.EMPLOYEES
    WHERE DEPARTMENT_ID = D.DEPARTMENT_ID AND EMPLOYEE_ID != M.EMPLOYEE_ID
)
ORDER BY D.DEPARTMENT_ID;
```

Problem 4:

Find the employees who get the second highest salary in their respective department. Show their first name, last name, department name, salary. The output should be like the table below:

FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_NAME
Matthew	Weiss	8000	Shipping
William	Gietz	8300	Accounting
...	...	...	...

Solution:

```
SELECT e.FIRST_NAME, e.LAST_NAME, e.SALARY, d.DEPARTMENT_NAME
FROM EMPLOYEES e
JOIN DEPARTMENTS d ON e.DEPARTMENT_ID = d.DEPARTMENT_ID
JOIN
(
      SELECT e.DEPARTMENT_ID, MAX(SALARY) SECOND_MAX_SALARY
      FROM EMPLOYEES e
      WHERE e.SALARY !=
      (
            SELECT MAX(e2.SALARY)
            FROM EMPLOYEES e2
            GROUP BY e2.DEPARTMENT_ID
            HAVING e2.DEPARTMENT_ID = e.DEPARTMENT_ID
      )
      GROUP BY e.DEPARTMENT_ID
) SECOND_TABLE
ON e.DEPARTMENT_ID = SECOND_TABLE.DEPARTMENT_ID
WHERE e.SALARY = SECOND_TABLE.SECOND_MAX_SALARY
```

Result has 9 rows

Problem 5:

For each employee, find out the number of unique posts he has served so far, including his current job. Sort the result by employee id. You cannot use any **AND**, **OR**, **NOT** operators. Use set operations instead. Sample output:

EMPLOYEE_ID	N_POSTS
100	1
101	3
...	...

Solution:

```
SELECT EMPLOYEE_ID, COUNT(*) N_POSTS
FROM (
    SELECT EMPLOYEE_ID, JOB_ID FROM hr.EMPLOYEES
    UNION
    SELECT EMPLOYEE_ID, JOB_ID FROM hr.JOB_HISTORY
)
GROUP BY EMPLOYEE_ID
ORDER BY EMPLOYEE_ID;
```

Result has 107 rows

Online 2 (Advanced SQL)

Section: A

Duration: 1 hour 20 minutes

Problem 1:

Print information of the employees that get the highest salary in his own department. Use **join** (50% marks will be deducted if done using subquery). Order the result by employee id. *Hint: while joining with other employees with higher salary, an employee with the highest salary will not find any other employee to join with.* Sample output:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	...	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	...	-	90
103	Alexander	Hunold	...	102	60
...	...	...	...	...	...
204	Hermann	Baer	...	101	70
205	Shelley	Higgins	...	101	110

12 rows

Solution:

```
SELECT DISTINCT E.*
FROM HR.EMPLOYEES E
    LEFT JOIN HR.EMPLOYEES T
    ON E.DEPARTMENT_ID = T.DEPARTMENT_ID
      AND E.SALARY < T.SALARY
WHERE T.EMPLOYEE_ID IS NULL
ORDER BY E.EMPLOYEE_ID;
```

Problem 2:

Print information of the employees that receive the highest salary under their managers. Order the results by employee id. Use **subquery** (50% marks will be deducted if done using join). Sample output:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	...	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	...	-	90
101	Neena	Kochhar	...	100	90
102	Lex	De Haan	...	100	90
...	...	...	...	...	...

22 rows

Solution:

```
SELECT *
FROM HR.EMPLOYEES E
WHERE NOT EXISTS (
    SELECT *
    FROM HR.EMPLOYEES
    WHERE SALARY > E.SALARY AND MANAGER_ID = E.MANAGER_ID
)
ORDER BY E.EMPLOYEE_ID;
```

Problem 3:

For each country, find the department with the highest salaried manager among the managers of all the departments located in that country. For example, consider the following one row of sample output. It implies that the manager of the Executive department has the highest salary among all the managers of the departments located in the United States of America. Sort the result by the country name.

COUNTRY_NAME	DEPARTMENT_NAME	SALARY
United States of America	Executive	24000

Solution:

```
SELECT C.COUNTRY_NAME, D.DEPARTMENT_NAME, M.SALARY
FROM HR.COUNTRIES C
    JOIN HR.LOCATIONS L ON C.COUNTRY_ID = L.COUNTRY_ID
    JOIN HR.DEPARTMENTS D USING (LOCATION_ID)
    JOIN HR.EMPLOYEES M ON D.MANAGER_ID = M.EMPLOYEE_ID
WHERE M.SALARY = (
    SELECT MAX(M2.SALARY)
    FROM HR.LOCATIONS L2
      JOIN HR.DEPARTMENTS D2 USING (LOCATION_ID)
      JOIN HR.EMPLOYEES M2 ON D2.MANAGER_ID = M2.EMPLOYEE_ID
    WHERE L2.COUNTRY_ID = C.COUNTRY_ID
)
ORDER BY C.COUNTRY_NAME;
```

Problem 4:

Write a query to retrieve the employees who have a salary greater than the average salary in their respective department.
Show their first name, last name, department name, salary and average salary of the department.
The column names will be like the table below:

FIRST_NAME	LAST_NAME	DEPARTMENT_NAME	SALARY	AVG_DEPARTMENT_SALARY
Matthew	Weiss	Shipping	8000	3475.555556
Adam	Fripp	Shipping	8200	3475.555556
Payam	Kaufling	Shipping	7900	3475.555556

Solution 1:

```
SELECT e.FIRST_NAME, e.LAST_NAME, d.DEPARTMENT_NAME, e.SALARY,
f.AVG_DEPARTMENT_SALARY
FROM EMPLOYEES e
JOIN DEPARTMENTS d ON e.DEPARTMENT_ID = d.DEPARTMENT_ID
JOIN
(
        SELECT DEPARTMENT_ID, AVG(e.SALARY) as AVG_DEPARTMENT_SALARY
        FROM EMPLOYEES e
        GROUP BY e.DEPARTMENT_ID
) f
ON e.DEPARTMENT_ID = f.DEPARTMENT_ID
WHERE e.SALARY > f.AVG_DEPARTMENT_SALARY
```

Solution 2:

```
SELECT e.FIRST_NAME, e.LAST_NAME, d.DEPARTMENT_NAME, e.SALARY,
(
        SELECT AVG(e2.SALARY)
        FROM EMPLOYEES e2
        GROUP BY DEPARTMENT_ID
        HAVING DEPARTMENT_ID = e.DEPARTMENT_ID
) as AVG_DEPARTMENT_SALARY
FROM EMPLOYEES e
JOIN DEPARTMENTS d ON d.DEPARTMENT_ID = e.DEPARTMENT_ID
WHERE e.SALARY >
(
        SELECT AVG(e2.SALARY)
        FROM EMPLOYEES e2
        GROUP BY DEPARTMENT_ID
        HAVING DEPARTMENT_ID = e.DEPARTMENT_ID
)
```

Result has 38 rows

Problem 5:

FInd out employees who are not managers and who do not have any prior experiences (experiences are stored in job history). Print the employee ids only. You cannot use any **NOT** operator. Use set operations instead.

Solution:

```
SELECT EMPLOYEE_ID
FROM HR.EMPLOYEES
MINUS
SELECT MANAGER_ID
FROM HR.EMPLOYEES
WHERE NVL(MANAGER_ID, -1) > -1
MINUS
SELECT EMPLOYEE_ID
FROM HR.JOB_HISTORY;
```

Result has 87 rows

Online 2 (Advanced SQL)

Section: B

Duration: 1 hour 20 minutes

Problem 1:

For all jobs, print the job id, job title and the number of managers (employees that manage other employees) doing that job. Order the result by job id. Use **join** (50% marks will be deducted if done using subquery). Sample output:

JOB_ID	JOB_TITLE	N_MANAGERES
AC_ACCOUNT	Public Accountant	0
AC_MGR	Accounting Manager	1
...	...	...

19 rows

Solution:

```
SELECT J.JOB_ID, J.JOB_TITLE, COUNT(DISTINCT M.EMPLOYEE_ID) AS N_MANAGERES
FROM HR.EMPLOYEES M
    JOIN HR.EMPLOYEES E ON E.MANAGER_ID = M.EMPLOYEE_ID
    RIGHT JOIN HR.JOBS J ON J.JOB_ID = M.JOB_ID
GROUP BY J.JOB_ID, J.JOB_TITLE
ORDER BY J.JOB_ID;
```

Problem 2:

Print information of the jobs that have the maximum salary range (salary range = max salary - min salary). Use **subquery** (50% marks will be deducted if done using join). Sample output:

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
AD_PRES	President	20080	40000

Solution:

```
SELECT *
FROM HR.JOBS
WHERE MAX_SALARY - MIN_SALARY = (
    SELECT MAX(MAX_SALARY - MIN_SALARY)
    FROM HR.JOBS
);
```

Problem 3:

For each department, find the ratio of the number of employees receiving above or equal to average salary and the number of employees receiving below or equal to average salary in that department. For example, consider the following one row of sample output. It shows that the Purchasing department has the average salary 4150, and the number of employees receiving salary below or equal to 4150 (the average salary of the purchasing department) is five times the number of employees receiving salary above or equal to 4150. In the output, omit the departments which do not have any employees. Sort the result by department id.

DEPARTMENT_ID	DEPARTMENT_NAME	AVG_SALARY	RATIO
30	Purchasing	4150	.2

Solution:

```
SELECT D.DEPARTMENT_ID, D.DEPARTMENT_NAME, T.AVG_SALARY, (
   SELECT COUNT(*)
   FROM HR.EMPLOYEES E
   WHERE E.DEPARTMENT_ID = D.DEPARTMENT_ID AND E.SALARY >= T.AVG_SALARY
  ) / (
   SELECT COUNT(*)
   FROM HR.EMPLOYEES E
   WHERE E.DEPARTMENT_ID = D.DEPARTMENT_ID AND E.SALARY <= T.AVG_SALARY
  ) AS RATIO
FROM HR.DEPARTMENTS D JOIN (
  SELECT DEPARTMENT_ID, AVG(SALARY) AS AVG_SALARY
  FROM HR.EMPLOYEES
  GROUP BY DEPARTMENT_ID
) T ON D.DEPARTMENT_ID = T.DEPARTMENT_ID
ORDER BY D.DEPARTMENT_ID;
```

Problem 4:

Find the employees who get the highest salary in their country. You can ignore the countries that do not have any employees.
Show their first name, last name, department name, salary and country name. The output should be like the table below:

FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_NAME	COUNTRY_NAME
Peter	Tucker	10000	Sales	Germany
Janette	King	10000	Sales	Germany
...	...	...	...	...

Solution:

```
SELECT e.FIRST_NAME, e.LAST_NAME, e.SALARY, d.DEPARTMENT_NAME, c.COUNTRY_NAME
FROM EMPLOYEES e
JOIN DEPARTMENTS d ON e.DEPARTMENT_ID = d.DEPARTMENT_ID
JOIN LOCATIONS l ON d.LOCATION_ID = l.LOCATION_ID
JOIN COUNTRIES c ON l.COUNTRY_ID = l.COUNTRY_ID
JOIN
(
      SELECT c.COUNTRY_ID, MAX(e.SALARY) AS MAX_SALARY
      FROM COUNTRIES c
      JOIN LOCATIONS l ON l.COUNTRY_ID = c.COUNTRY_ID
      JOIN DEPARTMENTS d ON d.LOCATION_ID = l.LOCATION_ID
      JOIN EMPLOYEES e ON e.DEPARTMENT_ID = d.DEPARTMENT_ID
      GROUP BY c.COUNTRY_ID
) COUNTRY_MAX_SALARY
ON c.COUNTRY_ID = COUNTRY_MAX_SALARY.COUNTRY_ID
WHERE e.SALARY = COUNTRY_MAX_SALARY.MAX_SALARY
```

Result has 7 rows

Problem 5:

Print employee id, salary and experience (in years) of those managers who either has experience of more than 18 years or salary greater than 10000. You cannot use any **AND**, **OR**, **NOT** operators. Use set operations instead. Sort the result by employee id. Sample output:

EMPLOYEE_ID	SALARY	EXPERIENCE
100	24000	20.58
101	17000	18.32
...	...	...

Solution:

```
SELECT
      EMPLOYEE_ID,
      SALARY,
      ROUND(MONTHS_BETWEEN(SYSDATE, HIRE_DATE) / 12, 2) AS EXPERIENCE
FROM HR.EMPLOYEES
WHERE EMPLOYEE_ID IN (
    SELECT MANAGER_ID
    FROM HR.EMPLOYEES
    INTERSECT (
        SELECT EMPLOYEE_ID
        FROM HR.EMPLOYEES
        WHERE MONTHS_BETWEEN(SYSDATE, HIRE_DATE) / 12 >= 18
        UNION
        SELECT EMPLOYEE_ID
        FROM HR.EMPLOYEES
        WHERE SALARY > 10000
    )
)
ORDER BY EMPLOYEE_ID;
```

Result has 17 rows