-- 1. Write a Query that get the date of the Third day in the next month

-- Print it in format like 14-december-2020, Saturday

select TO\_CHAR(DATE\_TRUNC('month', NOW()) + INTERVAL '1 month' + INTERVAL '2 days' , 'DD-Month-YYYY, Day') AS third\_day\_next\_month;

-- 2. Write a Query that get the last day date of the current month from today

-- Print it in format like 14-december-2020, Saturday

select TO\_CHAR(DATE\_TRUNC('month', NOW()) + INTERVAL '1 month' - INTERVAL '1 days' , 'DD-Month-YYYY, Day') AS Last\_day\_current\_month;

-- 3. Display the employee’s name, hire date and salary review date,

-- The salary review date is the day after six months and Five days of service.Label the column Review.

-- Format the dates to appear in the format similar to “Sunday, the 7th of September, 1981 “.

select first\_name,hire\_date,salary ,

TO\_CHAR(hire\_date + INTERVAL '6 month' + INTERVAL '5 days' , 'FMDay, the dth "of" FMMonth ,YYYY') AS salary\_review\_date

from hr.employees

-- 4. Write a query that will display the difference between the highest and lowest salaries in each department.

select department\_id, max(salary),min(salary),max(salary)-min(salary) as "difference"

from hr.employees

GROUP by department\_id

--correlated way

SELECT DISTINCT

e.department\_id,

(SELECT MAX(salary) FROM hr.employees WHERE department\_id = e.department\_id) -

(SELECT MIN(salary) FROM hr.employees WHERE department\_id = e.department\_id) AS difference

FROM

hr.employees e;

-- 5. write a query that will display the city,

-- department name number of employees and the average salary for all employee in that department,

--round the average salary to two decimal places.

SELECT l.city,

d.department\_name,

COUNT(e.employee\_id) AS num\_employees,

ROUND(AVG(e.salary), 2) AS avg\_salary

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

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

-- 6. Display the employee number, name and salary

--for all employee who earn more than the average salary in his department

--correlated way

select e.employee\_id, e.first\_name, e.salary

from hr.employees e

where e.salary > (select AVG(e2.salary) from hr.employees e2 where e2.department\_id = e.department\_id);

-- 7. Show Employees data Whose Salary is Higher Than Their Manager's

-- and show Manager name, salary ( use sub query not join )

SELECT e.employee\_id,

e.first\_name AS employee\_name,

e.salary AS employee\_salary,

(SELECT m.first\_name

FROM hr.employees m

WHERE m.employee\_id = e.manager\_id) AS manager\_name,

(SELECT m.salary

FROM hr.employees m

WHERE m.employee\_id = e.manager\_id) AS manager\_salary

FROM hr.employees e

WHERE e.salary > (

SELECT m.salary

FROM hr.employees m

WHERE m.employee\_id = e.manager\_id

);

-- 8. Show Employees data Who Earn the Lowest Salary in Their Department ( use subquery not join )

select e.employee\_id, e.first\_name, e.salary

from hr.employees e

where e.salary = (select min(e2.salary) from hr.employees e2 where e2.department\_id = e.department\_id);

-- 9. Find employees who have been hired earlier than anyone else in the same job ( use subquery not join )

select e.employee\_id, e.first\_name, e.salary , e.hire\_date

from hr.employees e

where e.hire\_date = (select min(e2.hire\_date) from hr.employees e2 where e2.job\_id = e.job\_id);

-- 10. Write a query to display employee\_id, last\_name, salary, dept id, dept name, job Id, job title, city, street address, country id, country name, region id, region name for all employees including those employees whose have no department too.

SELECT e.employee\_id,

e.last\_name,e.salary,e.department\_id,e.job\_id,

d.department\_name,

j.job\_title,

l.city,l.street\_address,

c.country\_id,c.country\_name,

r.region\_id,r.region\_name

FROM hr.employees e

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

LEFT JOIN hr.jobs j ON e.job\_id = j.job\_id

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

LEFT JOIN hr.countries c ON l.country\_id = c.country\_id

LEFT JOIN hr.regions r ON c.region\_id = r.region\_id;