

Modul Praktikum 6 - Oracle

Tujuan :

1. Mampu menggunakan Oracle Application Express (APEX)
2. Mampu membuat Query di APEX

Buatlah query berikut di APEX, screenshot hasilnya dan jelaskan maksud dari query tersebut!

1.

```
SELECT first_name, last_name, job_id, job_title
FROM employees NATURAL JOIN jobs
WHERE department_id > 80;
```
2.

```
SELECT last_name, department_name
FROM employees CROSS JOIN
departments;
```
3.

```
SELECT first_name, last_name, department_id, department_name
FROM employees JOIN departments USING (department_id);
```
4.

```
SELECT last_name, job_title
FROM employees e JOIN jobs j
ON (e.job_id = j.job_id);
```
5.

```
SELECT last_name, salary, grade_level, lowest_sal,
highest_sal
FROM employees JOIN job_grades
ON(salary BETWEEN lowest_sal AND highest_sal);
```
6.

```
SELECT last_name, department_name AS "Department", city
FROM employees JOIN departments USING (department_id)
JOIN locations USING (location_id);
```
7.

```
SELECT e.last_name, d.department_id,
d.department_name
FROM employees e LEFT OUTER JOIN
departments d
ON (e.department_id =
d.department_id);
```
8.

```
SELECT e.last_name, d.department_id,
d.department_name
FROM employees e RIGHT OUTER JOIN
departments d
ON (e.department_id =
d.department_id);
```

9.

```
SELECT e.last_name, d.department_id, d.department_name
FROM employees e FULL OUTER JOIN departments d
ON (e.department_id = d.department_id);
```
10.

```
SELECT last_name, e.job_id AS "Job", jh.job_id AS "Old job",
end_date
FROM employees e LEFT OUTER JOIN job_history jh
ON(e.employee_id = jh.employee_id);
```
11.

```
SELECT worker.last_name || ' works for ' || manager.last_name
AS "Works for"
FROM employees worker JOIN employees manager
ON (worker.manager_id = manager.employee_id);
```
12.

```
SELECT employee_id, last_name, job_id, manager_id
FROM employees
START WITH employee_id = 100
CONNECT BY PRIOR employee_id = manager_id
```
13.

```
SELECT last_name || ' reports to ' || PRIOR last_name AS "Walk
Top Down"
FROM employees
START WITH last_name = 'King'
CONNECT BY PRIOR employee_id = manager_id;
```
14.

```
SELECT LEVEL, last_name ||
' reports to ' ||
PRIOR last_name
AS "Walk Top Down"
FROM employees
START WITH last_name = 'King'
CONNECT BY PRIOR
employee_id = manager_id;
```
15.

```
SELECT LPAD(last_name, LENGTH(last_name)+
(LEVEL*2)-2, '_') AS "Org_Chart"
FROM employees
START WITH last_name = 'King'
CONNECT BY PRIOR employee_id = manager_id;
```
16.

```
SELECT last_name
FROM employees
WHERE last_name != 'Higgins'
START WITH last_name = 'Kochhar'
CONNECT BY PRIOR employee_id = manager_id;
```
17.

```
SELECT last_name, e.job_id, job_title
FROM employees e, jobs j
WHERE e.job_id = j.job_id
AND department_id = 80;
```
18.

```
SELECT employees.last_name, departments.department_name
FROM employees, departments;
```
19.

```
SELECT last_name, city
FROM employees e, departments d, locations l
WHERE e.department_id = d.department_id
AND d.location_id = l.location_id;
```

20.

```
SELECT last_name, salary, grade_level, lowest_sal,
highest_sal
FROM employees, job_grades
WHERE (salary BETWEEN lowest_sal AND highest_sal);
```
21.

```
SELECT e.last_name,
d.department_id,
d.department_name
FROM employees e, departments d
WHERE e.department_id =
d.department_id(+);
```
22.

```
SELECT e.last_name,
d.department_id,
d.department_name
FROM employees e, departments d
WHERE e.department_id(+) =
d.department_id;
```
23.

```
SELECT MAX(salary), MIN(salary), MIN(employee_id)
FROM employees
WHERE department_id = 60;
```
24.

```
SELECT COUNT(*)
FROM employees
WHERE hire_date < '01-Jan-1996';
```
25.

```
SELECT SUM(salary)
FROM employees
WHERE department_id = 90;
```
26.

```
SELECT SUM(DISTINCT salary)
FROM employees
WHERE department_id = 90;
```
27.

```
SELECT AVG(commission_pct)
FROM employees;
```
28.

```
SELECT AVG(NVL(commission_pct, 0))
FROM employees;
```
29.

```
SELECT department_id, AVG(salary)
FROM employees
GROUP BY department_id
ORDER BY department_id;
```
30.

```
SELECT department_id, job_id,
count(*)
FROM employees
WHERE department_id > 40
GROUP BY department_id, job_id;
```
31.

```
SELECT department_id, MAX(salary)
FROM employees
GROUP BY department_id
HAVING COUNT(*) > 1
ORDER BY department_id;
```

32.

```
SELECT department_id, job_id, SUM(salary)
FROM employees
WHERE department_id < 50
GROUP BY ROLLUP (department_id, job_id);
```
33.

```
SELECT department_id, job_id, SUM(salary)
FROM employees
WHERE department_id < 50
GROUP BY CUBE (department_id, job_id);
```
34.

```
SELECT department_id, job_id, manager_id, SUM(salary)
FROM employees
WHERE department_id < 50
GROUP BY GROUPING SETS
((job_id, manager_id), (department_id, job_id),
(department_id, manager_id));
```
35.

```
SELECT hire_date, employee_id, job_id
FROM employees
UNION
SELECT TO_DATE(NULL), employee_id,
job_id
FROM job_history;
```
36.

```
SELECT last_name, job_id, salary, department_id
FROM employees
WHERE job_id =
    (SELECT job_id
     FROM employees
     WHERE employee_id = 141)
AND department_id =
    (SELECT department_id
     FROM departments
     WHERE location_id = 1500);
```
37.

```
SELECT department_id, MIN(salary)
FROM employees
GROUP BY department_id
HAVING MIN(salary) >
    (SELECT MIN(salary)
     FROM employees
     WHERE department_id = 50);
```
38.

```
SELECT last_name, hire_date
FROM employees
WHERE EXTRACT(YEAR FROM hire_date) IN
    (SELECT EXTRACT(YEAR FROM hire_date)
     FROM employees
     WHERE department_id=90);
```
39.

```
SELECT last_name, hire_date
FROM employees
WHERE EXTRACT(YEAR FROM hire_date) < ANY
    (SELECT EXTRACT(YEAR FROM hire_date)
     FROM employees
     WHERE department_id=90);
```
40.

```
SELECT last_name, hire_date FROM employees
WHERE EXTRACT(YEAR FROM hire_date) < ALL
    (SELECT EXTRACT(YEAR FROM hire_date)
     FROM employees
     WHERE department_id=90);
```

41.

```
SELECT  employee_id,
        manager_id,
        department_id
FROM    employees
WHERE   manager_id IN
        (SELECT  manager_id
         FROM    employees
         WHERE   employee_id IN
              (149,174))
AND     department_id IN
        (SELECT  department_id
         FROM    employees
         WHERE   employee_id IN
              (149,174))
AND employee_id NOT IN(149,174);
```

42.

```
SELECT o.first_name,
       o.last_name, o.salary
FROM employees o
WHERE o.salary >
      (SELECT AVG(i.salary)
       FROM employees i
       WHERE i.department_id =
            o.department_id);
```

43.

```
SELECT last_name AS "Not a Manager"
FROM   employees emp
WHERE  NOT EXISTS
      (SELECT *
       FROM employees mgr
       WHERE mgr.manager_id = emp.employee_id);
```

44.

```
WITH managers AS
  (SELECT DISTINCT manager_id
   FROM employees
   WHERE manager_id IS NOT NULL)

SELECT last_name AS "Not a manager"
FROM employees
WHERE employee_id NOT IN
      (SELECT *
       FROM managers);
```