* SQL Queries on Multiple tables -

at least 1 attribute should common in both tables to call both tables or join tables

- Consider follo. database :
- Employee(empid, fname, lname, email, phone no, jobid, mid, date, salary, commision, deptid)
- Department(deptid, dname, mid, loc.id)
- Location (loc-id, street, pincode, city, state, cid).

(1) Write a query to display dept no, last name & dept name for all employees.

```
SQL> select Employee.deptid, lname, dname from
            Employee, Department
     where Employee.deptid = Department.deptid; ↵
```

or

```
SQL> select e.deptid, lname, dname
     from e Employee, d Department
     where e.deptid = d.deptid ; ↵
```

job-id = 20

create a unique listing of all jobs that are in department 80 include location of department in output.

id

```
select distinct jobid, loc_id
from Employee, Department
where ~~deptid~~ Department.deptid = 80
      and
   Department.deptid = Employee.deptid; ↵
```

Write a query to display lastname, dept name, loc_id & city of all employee who earn commision.

```
→ select lname, dname, d.loc_id, city
  from Employee, d Department, Location
  where Employee.dept_id = d.dept_id
        and
        ~~Do d.dept_id = Location.loc~~
        d.loc_id = Location.loc_id
        and
        commision is not null; ↵
```

4) Display employee last name, dept name of all employees who have 'a' in theire last name.

```
→ select lname, dname
  from Employee, Department
  where Employee.dept_id = Department.dept_id
      and
      lname like '%a%'; ↵
```

⑤ Write a query to display no. of people with same job allow with its jobid.

```
SQL> select count (jobid), jobid
     from Employee
     group by jobid ; ↵
```

⑥ Determine the no. of managers without listing them. Label the column with "Number of managers".

```
SQL> select count(distinct mid) "Number of managers"
     from Employee ; ↵
```

⑦ Write a query that displays the difference between highest & lowest ~~quat~~ salary & label the column as "difference."

```
                        ↓minus sign
SQL> select max(salary) - min(salary) "Difference"
     from Employee ;
```

⑧ Display the manager no. & lowest salary of all employee. Exclude whose manager id is not known. exclude any groups where minimum salary is 6000 or less.

```
SQL> select mid, min(salary)
     from Employee
     where mid is not null
     group by mid
     having min(salary) > 6000; ↵
```

⑨ Create a query to display emp no. & last name of all employees who earn more than average salary. Sort the result in ascending order by salary.

```
select empid, lname
from Employee
where salary > (select avg(salary)
                from Employee
               )
order by salary ; ↵
```

Write a query that displays the employee number & last name of employees who work in a department with any employee whose last name contains u.

```
select empid, lname
 from Employee
 where lname in (select lname from employee
                 where lname like '%u%');
```

Display the lastname, department no. & job id of all employees whose department location id is 170.

```
11) select lname, d.deptid, jobid
     from e Employee, d Department
     where e.deptid = d.deptid
        and
        loc-id =170 ; ↵
```

(12) Display employee no, last name and salary of all employees who earn more than average salary and who work in a department with 'u' in their last name.

```
SQL> select emp_id, lname, salary
     from Employee
     where salary > (select avg(salary)
                          from Employee)
      and
           lname like '%u%'; ↵
```

(13) List department id from department that do not contain jobid 20. using set operation.

```
SQL> select dept_id
     from Department
     ~~where~~. minus select dept_id
                 from Employee
          where jobid = 20;
```

(14) Produce a list of jobs for department 10, 15, 20 display jobid and dept_id using set operations.

```
SQL> select jobid, ~~dept_id~~ ~~Department~~ deptid
     from Employee~~, Department~~
     intersect select deptid
          from Department
     where deptid not in (10,15,20);
```

(15) Create a query to display the employee last name, jobid & salary who earn a salary higher than all employee with jobid 3001 sort the result highest to lowest with respect to salary.

```
SQL> select lname, jobid, salary
     from Employee.
     where salary > all (select salary
                         from Employee
                         where
```