NOTE: If you try the following queries in the sample database, you will get "no rows selected" for some of the queries because there are no data that matches the condition(s). The answer given is not wrong. Try with a different data value to see the result.

1.Show the manager for each department, include relevant information.

```
select manager_id, department_id, department_name
from departments.
```

2.Show all employees with the first name ‘Alexander’.

```
select employee_id, first_name, last_name, email
from employees
where first_name LIKE 'Alexander% ';

where first_name = 'Alexander';
```

Commented [JN1]: partial string matching “Alexanderian", "Alexander123"

3.Show all employees belonging to department 170.

```
select employee_id, first_name, last_name
from employees
where department_id = 170;
```

*Note:
There are no department 170, can try 110 (or others) instead.

4.Show all employees earning less than 10,000 belonging to department 170, ordering the result from the highest to the lowest salary.

```
select employee_id, first_name, last_name
from employees
where department_id = 170 and
salary < 10000
order by salary desc;
```

*Note:
There are no department 170, can try 110 (or others) instead.

5.Show all jobs that pay at least 15,000 salary.

```
Select job_id, job_title
from jobs
where min_salary >= 15000 OR
```

```
max_salary >=15000;
```

6.List all locations with a street address that has the lucky number “8” in it.

```
Select location_id, street_address, city
from locations
where street_address LIKE '%8%';
```

7.List all locations with a post code that ends with “18” or “28” or “98”.

```
Select location_id, street_address, postal_code, city
from locations
where (postal_code LIKE '%18') OR
      (postal_code LIKE '%28') OR
      (postal_code LIKE '%98');
```

8.List all employees that has been employed for between 5 to 8 years.

```
select employee_id, first_name, last_name, hire_date, job_id,
       Round((Months_Between(sysdate, hire_date)/12),0) as Working_Years
from employees
where (Months_Between(sysdate, hire_date)/12) BETWEEN 5 and 8
order by Working_Years;
```

Commented [JN2]: 0 means zero decimal places

Months_Between is a function

Note:
The query below will show that all employees have been working more than 10 years, change the 5 to 8 to something else to see results.

```
select employee_id, first_name, last_name, hire_date, job_id,
       (Months_Between(sysdate, hire_date)/12) as Working_Years
from employees
order by Working_Years;
```

Multiple table queries

9.List employees working in the state province of California.

```
select E.employee_id, E.last_name, D.department_name
from employees E, departments D, locations L
where E.department_id = D.department_id AND
D.location_id = L.location_id AND
state_province = 'California';
```

10.List employees working in countries beginning with the letter A.

```
select C.country_name, E.employee_id, E.last_name, D.department_name
from employees E, departments D, locations L, countries C
where E.department_id = D.department_id AND
D.location_id = L.location_id AND
L.country_id = C.country_id AND
country_name LIKE 'A%';
```

11.List all employees that had worked as a "SALES REPRESENTATIVE" previously (not including the current job).

```
select E.employee_id, E.last_name, E.first_name, JH.start_date, JH.end_date, J.job_title
from employees E, job_history JH, jobs J
where E.employee_id = JH.employee_id AND
JH.job_id = J.job_id AND
UPPER(J.job_title) = 'SALES REPRESENTATIVE';
```

Note: Remind students to read Chapter 5 of the SQL Language Reference on Functions; need to use functions in the Assignment

12.Show the salary details for the IT department.

```
select DISTINCT J.job_title, J.min_salary, J.max_salary
from departments D, employees E, jobs J
where E.job_id = J.job_id AND
E.department_id = d.department_id AND
D.department_name = 'IT'
```

Commented [JN3]: Distinct means Return unique rows

Note: Not many records for IT department, can try another department e.g. 'Marketing'

13.List all employees and their manager (must show manager's name)

```
select E.employee_id, E.first_name, E.last_name, E.manager_id, Mgr.first_name,
Mgr.last_name
from employees E, employees Mgr
where E.manager_id = Mgr.employee_id
order by E.employee_id;
```

list All MANAGERS and their subordinates - just change the sorting sequence:

```
select E.employee_id, E.first_name, E.last_name, E.manager_id, Mgr.first_name,
Mgr.last_name
```

```
from employees E, employees Mgr
where E.manager_id = Mgr.employee_id
order by Mgr.employee_id, E.employee_id;
```

14.List all employees that work in the same country.

```
select C.country_id, C.country_name, L.location_id, E.employee_id, E.first_name, E.last_name
from Countries C, Locations L, Departments D, Employees E
where (C.country_id  = L.country_id)     AND
      (L.location_id = D.location_id)    AND
      (D.department_id = E.department_id)
order by C.country_id, L.location_id;
```

*Note:
The above query doesn't work if there is ONE country with ONE location with ONE department having only ONE employee.
(i.e the ONLY employee working in a country, no colleagues...)

To exclude the ONE employee working in ONE country by himself:

```
select C.country_id, C.country_name, L.location_id, E.employee_id, E.first_name, E.last_name
from Countries C, Locations L, Departments D, Employees E
where (C.country_id  = L.country_id)      AND
      (L.location_id = D.location_id)     AND
      (D.department_id = E.department_id) AND
      (C.country_id NOT IN (select C.country_id
                           from Countries C, Locations L, Departments D, Employees E
                           where (C.country_id  = L.country_id)     AND
                                (L.location_id = D.location_id)    AND
                                (D.department_id = E.department_id)
                           group by C.country_id
                           having count(*) = 1))
order by C.country_id, L.location_id;
```

Commented [JN4]: used to group rows that have the same values in specified columns into summary rows, like "total" or "average" rows

Commented [JN5]: " * " representing all columns in a table

The nested select identifies countries that have only ONE employee.

15.How many employees had been a Stock Clerk previously?

```
Select COUNT(*) as No_Of_Stock_Clerk_Previously
from Jobs J, Job_History JH
where (J.job_id = JH.job_id)       AND
      (J.job_title = 'Stock Clerk');
```

16.How many current Stock Clerks are there?

```
Select COUNT(*) as No_Of_Stock_Clerk_Currently
from Jobs J, Employees E
where (J.job_id = E.job_id)       AND
      (J.job_title = 'Stock Clerk');
```

17.What is the total salary of all employees in the Marketing department?

```
select SUM(E.salary) as Total_Mktg_Dept_Salary
from Departments D, Employees E
where (D.department_id = E.department_id) AND
      (D.department_name = 'Marketing');
```

To see the raw data before SUM:

```
select E.salary
from Departments D, Employees E
where (D.department_id = E.department_id) AND
      (D.department_name = 'Marketing');
```

18.What is the average salary of all the Purchasing Clerk?

```
Select AVG(E.salary) as Avg_Purchasing_Clerk_Salary
from Jobs J, Employees E
where (J.job_id = E.job_id)       AND
      (J.job_title = 'Purchasing Clerk');
```

To see the raw data before AVG:

```
Select E.salary
from Jobs J, Employees E
where (J.job_id = E.job_id)       AND
      (J.job_title = 'Purchasing Clerk');
```

**NOTE:
There are 25 countries listed in the sample database but not all countries have departments. There are only a total of 27 departments located in Canada, Germany, United Kingdom and United States of America.

```
select C.country_name, D.Department_id, D.Department_name
from Countries C, Locations L, Departments D
where (C.country_id  = L.country_id)     AND
      (L.location_id = D.location_id)
```

```
order by C.country_name, D.Department_id;
```

If you or students would like to try to see result of your select statements, please change the country names for the queries below.

19.How many employees are there in Singapore?

```
select COUNT(*) as No_Of_Employees_In_Singapore
from Countries C, Locations L, Departments D, Employees E
where (C.country_name = 'Singapore')     AND
      (C.country_id  = L.country_id)     AND
      (L.location_id = D.location_id)    AND
      (D.department_id = E.department_id);
```

*Note: There are no departments in 'Singapore' in the database.

20.What is the total salary of each department located in Singapore?

```
select D.Department_id, D.Department_name, SUM(E.salary) as Total_Department_Salary
from Countries C, Locations L, Departments D, Employees E
where (C.country_name = 'Singapore')     AND
      ( _name;
```

*Note: There are no departments in 'Singapore' in the database.

21.What are the departments in Australia that have at least 5 employees?

```
select D.Department_id, D.Department_name, COUNT(*) as No_Of_Employees
from Countries C, Locations L, Departments D, Employees E
where (C.country_name = 'Australia')     AND
      (C.country_id  = L.country_id)     AND
      (L.location_id = D.location_id)    AND
      (D.department_id = E.department_id)
group by D.department_id, D.Department_name
having COUNT(*) >=5;
```

*Note: There are no departments in 'Australia' in the database, can try 'United Kingdom' or 'United States of America'

22.How many employees earn more salary than John Russel?

```
Select COUNT(*) as "Higher then John Russel"
from Employees E1, Employees Others
where (E1.first_name = 'John')   AND
      (E1.last_name  = 'Russel') AND
      (Others.salary > E1.salary);
```

*Note: There are no 'John Russel' in the database, can try other names to see a result. Example below uses employee 'Tayler Fox'

```
Select COUNT(*) as "Higher then Tayler Fox"
from Employees E1, Employees Others
where (E1.first_name = 'Tayler') AND
      (E1.last_name  = 'Fox')    AND
      (Others.salary > E1.salary);
```

To see the raw data for verification:

```
Select E1.first_name, E1.last_name, E1.Salary, Others.first_name, Others.last_name,
Others.salary
from Employees E1, Employees Others
where (E1.first_name = 'Tayler') AND
      (E1.last_name  = 'Fox')    AND
      (Others.salary >= E1.salary);
```

23. Identify the employee with the highest salary.

```
Select employee_id, first_name, Last_name, salary
from Employees
where salary =(Select max(salary)
                        from employees);
```

Note: There may be more than ONE employee with the same highest salary

24. Identify the employee for the job that has the highest salary. Include relevant information.

```
Select employee_id, first_name, Last_name, E.job_id, J.job_title, salary
from Employees E, Jobs J
where  E.job_id IN      (Select job_id
               from jobs
               where max_salary = (select max(max_salary)
                                 from jobs))
          AND
        E.job_id = J.job_id;
```

Note: it may be possible that more than one job_id can have the same highest max_salary, hence the use of the IN operator.
In the example below, there are two job_id with the maximum salary of 16,000.

```
Select employee_id, first_name, Last_name, E.job_id, J.job_title, salary
from Employees E, Jobs J
where  E.job_id IN      (Select job_id
               from jobs
               where max_salary = 16000)
          AND
        E.job_id = J.job_id;
```