Nested queries

1. Set operation (union, intersect, mines)

2. Set membership (IN, NOT IN)

3. Set comparison (< ,>,=, <>,<=,>=)

Joins

1. Inner join

2. Outer join: a. left

b. right

c. full

* with natural join alone = inner join
* natural join means that takes the common column/s as a one column when we execute the sql statement contains the natural statement and we don’t need using **ON** statement

Q0) Describe the result of each query below by its own language depend on HR Schema:

select \* from job\_history جدول العمل السابق

select \* from employeesجدول الموظفين

create view xx as

select e.employee\_id, e.first\_name, e.hire\_date, e.job\_id as Current\_job\_id, e.department\_id as Current\_department\_id, d.department\_name as Current\_department\_name, j.start\_date, j.end\_date, j.job\_id as Past\_job\_id, j.department\_id as Past\_department\_id

from employees e, job\_history j, departments d

where e.employee\_id = j.employee\_id and e.department\_id = d.department\_id

هنا تم انشاء فيو مكونة من 3 جداول حيث تم اختيار رقم الموظف والاسم الاول له وتاريخ التعيين بالشركة وكذلك المسمى الوظيفي الحالي ورقم القسم الحالي وهذا كله (من جدول الموظفين) وكذلك اسم القسم الحالي (من جدول الأقسام) وايضا تم اختيار بداية ونهاية تاريخ الوظيفة السابقة له بالشركة وكذلك المسمى الوظيفي له انذاك ورقم القسم اللذي كان يعمل به سابقا (وهذا كله من جدول العمل السابق) و تم انشاء ربط ما بين الثلاث جداول بطريقة ربط جدول الموظفين مع جدول تاريخ العمل بواسطة مفتاح رقم الموظف وتم ربط جدول الاقسام مع جدول الموظفين من خلال مفتاح رقم القسم ونتج لدينا جدول مخزن داخل فيو تحتوي على جميع العناصر اللتي تم ذكرها

select x.EMPLOYEE\_ID, x.FIRST\_NAME,x.hire\_date, x.Current\_job\_id, x.CURRENT\_DEPARTMENT\_ID, x.CURRENT\_DEPARTMENT\_NAME, x.start\_date, x.end\_date, x.Past\_job\_id,x.PAST\_DEPARTMENT\_ID, d.department\_name

from xx x, departments d

where x.PAST\_DEPARTMENT\_ID=d.department\_id

هنا تم ربط الفيو السابقة مع جدول الاقسام من اجل اظهار اسم اسم القسم السابق اللذي كان يعمل به الموظف حيث ان الفيو لم تظهر ذلك بسبب ارتباط جدول الاقسام مع جدول الموظفين فقط وليس مع جدولي (الموظفين وجدول العمل السابق)

select first\_name, extract(year from current\_date) - extract (year from hire\_date) as number\_of\_years\_in\_company

from employees

1. Write a query to display the name (first name and last name) for those employees who gets more salary than the employee whose ID is 163.

SELECT first\_name,last\_name

FROM employees

WHERE salary > (SELECT salary

FROM employees

WHERE employee\_id = 163)

1. Write a query to display the name (first name and last name), salary, department id, job id for those employees who works in the same designation (department\_id) as the employee works whose id is 169.
2. Write a query to display the name (first name and last name), salary, department id for those employees who earn such amount of salary which is the smallest salary of any of the departments.

SELECT first\_name,last\_name,salary,department\_id

FROM employees

WHERE salary IN (SELECT MIN(salary)

FROM employees

GROUP BY department\_id);

1. Write a query to display the employee id, employee name (first name and last name) for all employees who earn more than the average salary.
2. Write a query to display the employee name (first name and last name), employee id and salary of all employees who report to ‘Payam’.

SELECT first\_name, last\_name, employee\_id, salary

FROM employees

WHERE manager\_id = (SELECT employee\_id

FROM employees

WHERE first\_name = 'Payam');

1. Write a query to display the department number, name (first name and last name), job\_id and department name for all employees in the Finance department.
2. Display all the information of an employee whose id is any of the number 134, 159 and 183
3. Write a query to display all the information of the employees whose salary is within the range of smallest salary and 2500.

SELECT \*

FROM employees

WHERE salary BETWEEN (SELECT MIN(salary)

FROM employees) AND 2500.00;

1. Write a query to display the employee name (first name and last name) and hire date for all employees in the same department as Clara. Exclude Clara

SELECT first\_name, last\_name, hire\_date

FROM employees

WHERE department\_id = (SELECT department\_id

FROM employees

WHERE first\_name = 'Clara')

AND first\_name != 'Clara';

1. Write a query to display the employee number, name (first name and last name), and salary for all employees who earn more than the average salary and who work in a department with any employee with a J in their name.
2. Display the employee name (first name and last name), employee id, and job title for all employees whose department location is Toronto

select e.employee\_id, e.first\_name, e.last\_name, e.job\_id

from employees e, departments d, locations l

where e.department\_id = d.department\_id and d.location\_id=l.location\_id and l.city = 'Toronto'

1. Write a query in SQL to display the department code and name for all departments which located in the city London
2. Write a query in SQL to display the full name (first and last name) of manager who is supervising 4 or more

SELECT first\_name||' '||last\_name AS full\_name

FROM employees

WHERE employee\_id IN

(SELECT manager\_id, count(employee\_id)

FROM employees

GROUP BY manager\_id

having count(employee\_id) >= 4)

1. Write a query in SQL to display all the information about those employees who earn second lowest salary of all the employees

SELECT \*

FROM employees

WHERE salary IN (SELECT MIN(salary)

FROM employees

WHERE salary > (SELECT MIN(salary)

from employees)

1. Write a query in SQL to display first name, last name, department id and name for all employees and departments including those where does not have any employee.
2. Write a query in SQL to display the first name of all employees and the first name of their manager including those who does not working under any manager.

SELECT e1.first\_name AS "employee\_name",

e2.first\_name AS "manager\_name"

FROM employees e1

LEFT JOIN employees e2

ON e1.manager\_id = e2.employee\_id;

1. Write a query in SQL to display the name of the department, average salary and number of employees working in that department who got commission

SELECT d.department\_name,

AVG(e.salary),

COUNT(commission\_pct)

FROM employees e

JOIN departments d

ON e.department\_id = d.department\_id

GROUP BY d.department\_name;

1. Write a query in SQL to display the employee ID, job name, number of days worked in for all those jobs in department 80

SELECT jh.employee\_id,

j.job\_title,

(jh.end\_date - jh.start\_date) AS num\_days

FROM jobs j

INNER JOIN job\_history jh

ON j.job\_id = jh.job\_id

WHERE jh.department\_id=80

1. Write a query in SQL to display the department name and number of employees in each of the department

SELECT d.department\_name,

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

FROM departments d

left outer JOIN employees e

ON d.department\_id = e.department\_id

GROUP BY d.department\_name

1. Write a query in SQL to display the full name (first and last name) of employee with ID and name of the country presently where (s)he is working.

SELECT e.first\_name|| ' '|| e.last\_name AS full\_name, e.employee\_id, c.country\_name

FROM employees e INNER JOIN departments d

ON e.department\_id = d.department\_id

INNER JOIN locations l

ON d.location\_id = l.location\_id

INNER JOIN countries c

ON l.country\_id = c.country\_id

SELECT e.first\_name|| ' '|| e.last\_name AS full\_name, e.employee\_id, c.country\_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