**1.Create a SQL statement to list all managers and their titles**

Here I selected all the information I needed and used left join because I wanted people to appear that are in the manager table, and both tables, but not just in the employees table.

**SELECT e.first\_name, e.last\_name, t.title, dm.emp\_no**

**FROM dept\_manager dm**

**LEFT JOIN employees e**

**ON dm.emp\_no = e.emp\_no**

**LEFT JOIN titles t**

**ON e.emp\_no = t.emp\_no;**

A screenshot of a computer program

Description automatically generated

**2.Create a SQL statement to show the salary of all employees and their department name**

We have to join dept\_emp and dept\_manager.

**CREATE OR REPLACE TEMPORARY TABLE join\_emp\_manager**

**SELECT distinct subquery.\*  
FROM  
(SELECT e.emp\_no, de.dept\_no AS dept\_emp, dm.dept\_no AS dept\_man, e.first\_name, e.last\_name,  
(SELECT GROUP\_CONCAT(dept\_name) from departments**

**where dept\_no = de.dept\_no OR dept\_no = dm.dept\_no) as Department,**

**e.birth\_date, e.hire\_date, e.gender**

**FROM**

**employees e**

**left join dept\_manager dm**

**on dm.emp\_no = e.emp\_no**

**left join dept\_emp de**

**on de.emp\_no = e.emp\_no**

**left join departments d**

**on de.dept\_no = d.dept\_no**

**or dm.dept\_no = d.dept\_no)subquery;**

**SELECT j.emp\_no, j.first\_name,j.last\_name, GROUP\_CONCAT( DISTINCT j.Department) as Department, GROUP\_CONCAT(DISTINCT s.salary) as Salary  
from join\_emp\_manager j  
left join salaries s  
on j.emp\_no = s.emp\_no  
group by j.emp\_no;**

A screenshot of a computer

Description automatically generated

**3.Create a SQL statement to show the hire date and birth date who belongs to HR department**

It’s asking that for each employee in HR department, we show hire date and birth date. So we need to join employees with dept\_emp with departments

The HR department has No. d003. From dept\_emp table, we can see that emp\_no 10005 works in this department. From dept\_manager table, we can see that emp\_no 10011 and 10012 work in this department. So we have to join dept\_emp and dept\_manager.

**CREATE OR REPLACE TEMPORARY TABLE join\_emp\_manager2**

**SELECT distinct subquery.\*  
FROM  
(SELECT e.emp\_no, de.dept\_no AS dept\_emp, dm.dept\_no AS dept\_man, e.first\_name, e.last\_name,  
(SELECT GROUP\_CONCAT(dept\_name) from departments**

**where dept\_no = de.dept\_no OR dept\_no = dm.dept\_no) as Department,**

**e.birth\_date, e.hire\_date, e.gender**

**FROM**

**employees e**

**left join dept\_manager dm**

**on dm.emp\_no = e.emp\_no**

**left join dept\_emp de**

**on de.emp\_no = e.emp\_no**

**left join departments d**

**on de.dept\_no = d.dept\_no**

**or dm.dept\_no = d.dept\_no)subquery;**

**SELECT emp\_no, first\_name,last\_name, Department, birth\_date, hire\_date  
from join\_emp\_manager**

**WHERE Department = ‘human resources’;**

A screenshot of a computer

Description automatically generated

**4.Create a SQL statement to show all departments and their department’s managers.**

We need to join departments, dept\_manager, and employees.

**SELECT d.dept\_name, d.dept\_no, e.first\_name, e.last\_name**

**FROM departments d**

**LEFT JOIN dept\_manager dm**

**ON d.dept\_no = dm.dept\_no**

**LEFT JOIN employees e**

**ON dm.emp\_no = e.emp\_no**

**ORDER BY d.dept\_name;**

A screenshot of a computer

Description automatically generated

**5.Create a SQL statement to show a list of HR’s employees who were hired after 1986**

We can already see the answer from question 3 but I created another temporary table

**CREATE OR REPLACE TEMPORARY TABLE join\_emp\_manager3**

**SELECT distinct subquery.\*  
FROM  
(SELECT e.emp\_no, de.dept\_no AS dept\_emp, dm.dept\_no AS dept\_man, e.first\_name, e.last\_name,  
(SELECT GROUP\_CONCAT(dept\_name) from departments**

**where dept\_no = de.dept\_no OR dept\_no = dm.dept\_no) as Department,**

**e.birth\_date, e.hire\_date, e.gender**

**FROM**

**employees e**

**left join dept\_manager dm**

**on dm.emp\_no = e.emp\_no**

**left join dept\_emp de**

**on de.emp\_no = e.emp\_no**

**left join departments d**

**on de.dept\_no = d.dept\_no**

**or dm.dept\_no = d.dept\_no)subquery;**

**SELECT emp\_no, first\_name,last\_name, Department, birth\_date, hire\_date  
from join\_emp\_manager**

A screenshot of a computer

Description automatically generated**WHERE Department = ‘human resources’ AND hire\_date > 1986;**

**6.Create a SQL statement to increase any employee’s salary up to 2%. Assume the employee has just phoned in with his/her last name.**

Here I created a temporary table as I can’t change the original tables.

**CREATE OR REPLACE TEMPORARY TABLE new\_salary AS (**

**SELECT s.salary, e.emp\_no, e.first\_name, e.last\_name**

**FROM salaries s**

**INNER JOIN employees e**

**ON s.emp\_no = e.emp\_no);**

**DELIMITER $$**

**CREATE PROCEDURE UpdatedSalaryByLastName(IN input\_last\_name VARCHAR(16))**

**BEGIN**

**UPDATE new\_salary**

**SET salary = salary \* 1.02**

**WHERE last\_name = input\_last\_name;**

**END $$**

**CALL UpdatedSalaryByLastName( ‘Simmel’ );**

**SELECT s.salary, e.emp\_no, e.first\_name, e.last\_name**

**FROM salaries s**

**INNER JOIN employees e**

**ON s.emp\_no = e.emp\_no WHERE last\_name = ‘Simmel’;**

**7.Create a SQL statement to delete employee’s record who belongs to marketing department and name start with A**

Looking at the marketing department, the department id is d001. From dept\_emp, no one works in d001. From dept\_manager, 2 people work in d001 with emp\_no 10001 and 10013. These 2 are Georgi and Eberhardt, and their names don’t start with A. So no one needs to be deleted. But to verify this:

**CREATE TEMPORARY TABLE marketing\_employees AS (**

**SELECT e.first\_name, e.last\_name, e.emp\_no, d.dept\_name**

**FROM employees e**

**INNER JOIN**

**dept\_manager dm**

**ON e.emp\_no = dm.emp\_no**

**INNER JOIN departments d**

**ON dm.dept\_no = d.dept\_no**

**WHERE d.dept\_name = ‘Marketing’**

**);**

**DELETE FROM marketing\_employees**

**WHERE first\_name = “A%”;**

**SELECT \* FROM marketing\_employees;**

A close-up of a number

Description automatically generated

**8.Create a database view to list the full names of all departments’ managers, and their salaries.**

Here I joined the departments managers table with employees and salaries. I used LEFT JOIN as I wanted to include everyone in the dept\_manager table, and in both dept\_manager and employees table, but not just the employees table. I adde max to s.salary as georgi was showing up twice even though he’s been manager once, so we want his max salary as that’s when he was manager.

**CREATE OR REPLACE VIEW dept\_mng\_salaries AS**

**SELECT d.dept\_name, d.dept\_no, e.first\_name, e.last\_name, e.emp\_no, MAX(s.salary)**

**FROM departments d**

**RIGHT JOIN dept\_manager dm**

**ON d.dept\_no = dm.dept\_no**

**LEFT JOIN employees e**

**ON dm.emp\_no = e.emp\_no**

**LEFT JOIN salaries s**

**ON e.emp\_no = s.emp\_no**

**GROUP BY emp\_no;**

**SELECT \* FROM dept\_mng\_salaries;**

A screenshot of a computer

Description automatically generated

**9.Create a database view to list all departments and their department’s managers, who**

**were hired between 1980 and 1990.**

Here, I took into account not only the hire date, but also the from date, and if the from date was less than hire date, I replaced the hire date with the from\_date.

**CREATE VIEW Managers\_80\_to\_90 AS**

**WITH modified\_hire\_dates AS (**

**SELECT**

**e.emp\_no, e.first\_name, e.last\_name, e.hire\_date, dm.from\_date,**

**CASE**

**WHEN dm.from\_date < e.hire\_date THEN dm.from\_date**

**ELSE e.hire\_date**

**END**

**AS new\_hire\_date**

**FROM employees e**

**INNER JOIN dept\_manager dm ON e.emp\_no = dm.emp\_no)**

**SELECT first\_name, last\_name, emp\_no, dept\_name**

**CASE**

**WHEN new\_hire\_date < hire\_date THEN new\_hire\_date**

**ELSE hire\_date**

**END AS hire\_date**

**FROM modified\_hire\_dates;**

**SELECT \* FROM Managers\_80\_to\_90**

**WHERE YEAR(hire\_date) BETWEEN 1980 AND 1990;**

A screenshot of a computer code

Description automatically generated

But I also did the version assuming hire\_date is the hire\_date. This gives fewer results as in the table Saniya and Patricio aren't included as Saniya’s hire\_date – 1994 > from\_date – 1989. Also Patricio’s hire\_date – 1992 > from\_date – 1985.

**CREATE VIEW manage\_1980to1990 AS**

**SELECT d.dept\_name, d.dept\_no, e.first\_name, e.last\_name , e.emp\_no, e.hire\_date**

**FROM departments d**

**LEFT JOIN dept\_manager dm**

**ON d.dept\_no = dm.dept\_no**

**LEFT JOIN employees e**

**ON dm.emp\_no = e.emp\_no**

**WHERE YEAR(hire\_date) BETWEEN 1980 AND 1990;**

**SELECT \* FROM manage\_1980to1990;**

A close-up of a list

Description automatically generated

**10.Create a SQL statement to increase salaries of all department’s managers up to 10%**

**who are working since 1990.**

Here I created a temporary table as I can’t change the actual data.

**CREATE TEMPORARY TABLE up\_sal\_manage7 AS**

**(SELECT d.dept\_name, d.dept\_no, e.first\_name, e.last\_name AS dept\_manager, e.emp\_no, s.salary, e.hire\_date**

**FROM departments d**

**LEFT JOIN dept\_manager dm**

**ON d.dept\_no = dm.dept\_no**

**LEFT JOIN employees e**

**ON dm.emp\_no = e.emp\_no**

**LEFT JOIN salaries s**

**ON e.emp\_no = s.emp\_no**

**WHERE YEAR(hire\_date) >= 1990**

**ORDER BY dept\_name);**

**UPDATE up\_sal\_manage7 SET salary = salary \* 1.1;**

**SELECT \* FROM up\_sal\_manage7;**

A close-up of a white paper

Description automatically generated

Notably, saniyakalloufi’s from date is earlier than the hire date. Above I have assumed that the hire\_date is the actual hire\_date. I considered replacing the hire\_dates with from\_dates in the case of the from\_date being smaller than the hire\_date, but this would only result in Mary, who doesn’t get paid (doesn’t have a salary) so this wouldn’t work.