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

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

**FROM**

**(SELECT** dm**.**emp\_no**,** dm**.**dept\_no**,** d**.**dept\_name **FROM** dept\_manager dm

**JOIN** departments d **ON** dm**.**dept\_no **=** d**.**dept\_no**)** **AS** mdn

**LEFT** **JOIN** employees e **ON** mdn**.**emp\_no **=** e**.**emp\_no

**LEFT** **JOIN** titles t **ON** e**.**emp\_no **=** t**.**emp\_no **;**

I first wrote the subquery to get all the department managers and their employee numbers, as well as the name of the department that they belong to. Then I join the table (mdn) from the subquery onto the employees and titles tables so that I can SELECT the managers’ names and job titles.

# A computer screen shot of a black screen Description automatically generated

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

**CREATE** **TEMPORARY** **TABLE** emp\_dept\_sal**(**

**WITH** cte **AS**

**(SELECT** dm**.**emp\_no**,** dm**.**dept\_no**,** d**.**dept\_name**,** dm**.**from\_date**,** dm**.to\_date** **FROM** dept\_manager dm

**LEFT** **JOIN** departments d **ON** dm**.**dept\_no **=** d**.**dept\_no

**UNION**

**SELECT** de**.**emp\_no**,** de**.**dept\_no**,** d**.**dept\_name**,** de**.**from\_date**,** de**.to\_date** **FROM** dept\_emp de

**LEFT** **JOIN** departments d **ON** de**.**dept\_no **=** d**.**dept\_no**)**

**SELECT** cte**.**dept\_name**,** e**.**emp\_no**,** e**.**first\_name**,** e**.**last\_name**,** s**.**salary**,** s**.**from\_date **AS** sal\_from**,** s**.to\_date** **AS** sal\_to**,**

**ROW\_NUMBER()** **OVER(PARTITION** **BY** e**.**emp\_no**,** cte**.**dept\_name **ORDER** **BY** s**.**salary **DESC)** **AS** row\_num

**FROM** cte

**RIGHT** **JOIN** employees e **ON** cte**.**emp\_no **=** e**.**emp\_no

**LEFT** **JOIN** salaries s **ON** e**.**emp\_no **=** s**.**emp\_no

**AND** s**.**from\_date **>=** cte**.**from\_date

**)**

For this question I needed employees that are both in the managers table and the emp\_dept table in order to find their department. To put that combined information into one table (cte) I’m using a UNION. I’m then using a right join to join the cte to the employees table as I was the information about all the employees (some are missing from the dept\_man and emp\_dept tables) and joining that onto the salaries table. Within my SELECT statement I’m using a window function to give all employees a ROW\_NUMBER, partitioned by emp\_no and dept\_name so that if the employee has two different salaries within the same department, the second instance will have a ROW\_NUMBER of 2. I’m then joining the employees and salaries tables, using **AND** s**.**from\_date **>=** cte**.**from\_date salaries to make sure that I only get the most recent salaries as some employees have had several different salaries. All of this is then put into a table so that I can select from that table and filter out any redundant rows.

A screenshot of a computer screen

Description automatically generated

**SELECT** **\*** **FROM** emp\_dept\_sal

**EXCEPT**

**SELECT** **\*** **FROM** emp\_dept\_sal

**WHERE** row\_num **!=** 1 **AND** salary **IS** **NULL**

Here I am selecting from the new table and filtering out the row where an employee has a second entry in the same department but without a salary.

A screenshot of a computer

Description automatically generated

# 3. Create a SQL statement to show the hire date and birth date of employees who belong to HR department.

**SELECT** e.first\_name, e.last\_name, e**.**hire\_date**,** e**.**birth\_date **FROM** employees e

**LEFT** **JOIN** dept\_emp de **ON** e**.**emp\_no **=** de**.**emp\_no

**LEFT** **JOIN** departments d **ON** de**.**dept\_no **=** d**.**dept\_no

**WHERE** d**.**dept\_name **=** 'Human Resources'

**UNION**

**SELECT** e.first\_name, e.last\_name, e**.**hire\_date**,** e**.**birth\_date **FROM** employees e

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

**LEFT** **JOIN** departments d **ON** dm**.**dept\_no **=** d**.**dept\_no

**WHERE** d**.**dept\_name **=** 'Human Resources'

As some employees appear only in dept\_emp and some in dept\_manager, I am using a union to combine the two queries joinin those tables onto dept\_name and filtering to select only Human Resources employees.

A black screen with white text

Description automatically generated

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

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

**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**;**

I am selecting from the departments table and joinin on the dept\_manager and left joining employees tables to retrieve information only about the managers.

A screen shot of a computer

Description automatically generated

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

**SELECT** e**.**emp\_no**,** e**.**first\_name**,** e**.**last\_name**,** e**.**hire\_date **FROM** employees e

**LEFT** **JOIN** dept\_emp de **ON** e**.**emp\_no **=** de**.**emp\_no

**LEFT** **JOIN** departments d **ON** de**.**dept\_no **=** d**.**dept\_no

**WHERE** d**.**dept\_name **=** 'Human Resources' **AND** YEAR**(**e**.**hire\_date**)** **>=** 1986

**UNION**

**SELECT** e**.**emp\_no**,** e**.**first\_name**,** e**.**last\_name**,** e**.**hire\_date **FROM** employees e

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

**LEFT** **JOIN** departments d **ON** dm**.**dept\_no **=** d**.**dept\_no

**WHERE** d**.**dept\_name **=** 'Human Resources' **AND** YEAR**(**e**.**hire\_date**)** **>=** 1986**;**

I an using the same query from question 3, and filtering the employees on the hire date to make sure that they were hired after 1986.

A screen shot of a computer

Description automatically generated

# 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.

**CREATE** **TEMPORARY** **TABLE** emp\_sal **AS**

**SELECT** e**.\*,** s**.**salary**,** s**.**from\_date**,** s**.to\_date** **FROM** employees e

**LEFT** **JOIN** salaries s **On** e**.**emp\_no **=** s**.**emp\_no**;**

As an analyst I cannot UPDATE data on the original tables, so I created a Temp table containing all employees and their salaries.

A screenshot of a computer screen

Description automatically generated

**UPDATE** emp\_sal

**SET** salary **=**

**CASE** last\_name **WHEN** **~insert\_last\_name~** **THEN** salary **\*** 1.02

**END**

Using an UPDATE statement on the Temporary table, using the relevant employees last name in the CASE clause to increase their salary by 2%.

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

**CREATE** **TEMPORARY** **TABLE** emp\_dept\_full **AS**

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

**LEFT** **JOIN** dept\_emp de **ON** e**.**emp\_no **=** de**.**emp\_no

**LEFT** **JOIN** departments d **ON** de**.**dept\_no **=** d**.**dept\_no

**WHERE** dept\_name **IS** **NOT** **NULL**

**UNION**

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

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

**LEFT** **JOIN** departments d **ON** dm**.**dept\_no **=** d**.**dept\_no

**WHERE** dept\_name **IS** **NOT** **NULL;**

As I cannot DELETE from the original table I created a Temp Table, using the same query from question 3 to get department information about employees from both the dept\_emp and dept\_manager tables using a UNION.

A screenshot of a computer

Description automatically generated

**INSERT** **INTO** emp\_dept\_full **VALUES**

**(**99999**,** 'Andrew'**,** 'Anderson'**,** 'Marketing'**);**

I’ve then inserted into the Temp Table a row to satisfy the conditions of the question as there originally is no employee with first name starts with A and works in the Marketing department.

**DELETE** **FROM** emp\_dept\_full

**WHERE**

dept\_name **=** 'marketing' **AND** first\_name **LIKE** 'a%'**;**

And then I DELETE that record from the temporary table using a WHERE clause to eliminate that specific employee.

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

**CREATE** **VIEW** full\_dept\_man\_sal **AS**

**SELECT** dm**.**emp\_no**,** e**.**first\_name**,** e**.**last\_name**,** d**.**dept\_name**,** d**.**dept\_no**,** s**.**salary **FROM** employees e

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

**RIGHT** **JOIN** departments d **ON** dm**.**dept\_no **=** d**.**dept\_no

**LEFT** **JOIN** salaries s **ON** e**.**emp\_no **=** s**.**emp\_no**;**

I create a VIEW by LEFT joining the dept\_manger table so that I have all the employees, RIGHT joining departments so that I have all the departments and LEFT joinin the salaries.

*NOTE: There was some disagreement in the classroom about this question – to me, the question implies that I need the information about the managers of all the departments, rather than the information about the department managers, which is why I use a RIGHT join on departments, to make sure that all the departments are included as not all of them are mentioned in the dept\_manager table.*

A screen shot 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.

**CREATE** **VIEW** dept\_man\_hired\_between\_1980\_1990 **AS**

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

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

**RIGHT** **JOIN** departments d **ON** dm**.**dept\_no **=** d**.**dept\_no

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

Here I am RIGHT joining the departments table to employees and dept\_manager tables to make sure that all departments are mentioned, and filtering by hire date using BETWEEN.

A screen shot of a computer

Description automatically generated

# 10. Create a SQL statement to increase salaries of all department’s managers up to 10% who are working since 1990.

**CREATE** **TEMPORARY** **TABLE** dept\_man\_sal **AS**

**SELECT** dm**.**emp\_no**,** e**.**first\_name**,** e**.**last\_name**,** d**.**dept\_name**,** d**.**dept\_no**,** s**.**salary**,** e**.**hire\_date **FROM** employees e

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

**RIGHT** **JOIN** departments d **ON** dm**.**dept\_no **=** d**.**dept\_no

**LEFT** **JOIN** salaries s **ON** e**.**emp\_no **=** s**.**emp\_no**;**

The question asks for **all** department’s managers so I’m RIGHT joining the departments table to include all departments, and then joining salaries. I created a Temp Table so that I can UPDATE the salaries.

A screen shot of a computer

Description automatically generated

**UPDATE** dept\_man\_sal

**SET** salary **=**

**CASE** YEAR**(**hire\_date**)** **WHEN** **<=**1990 **THEN** salary **\*** 1.10

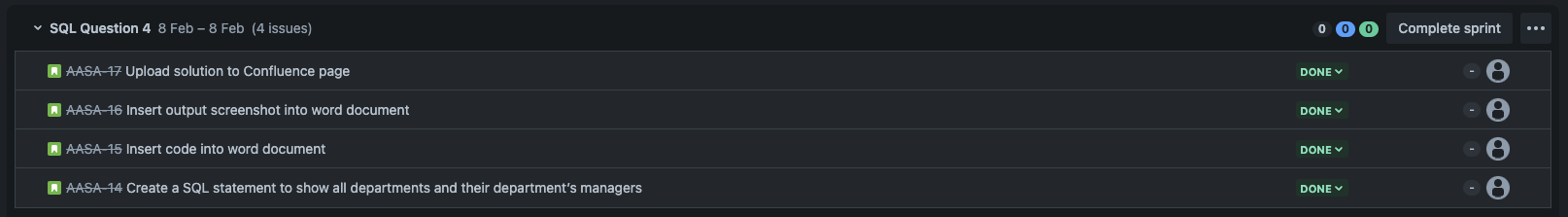
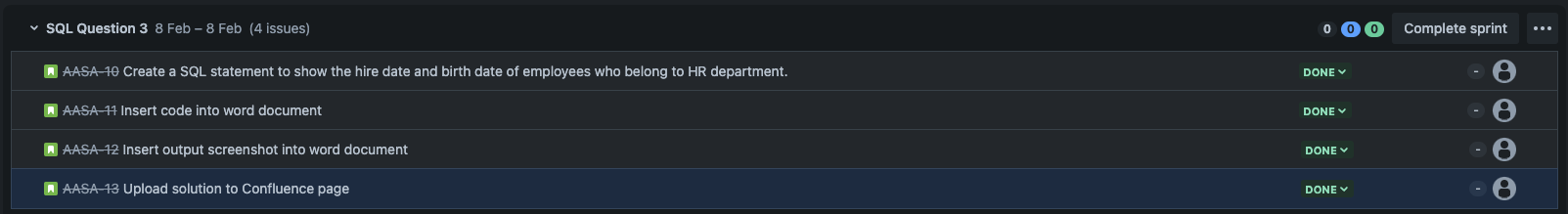
**END**

I’m using an UPDATE statement on the temporary table, and using CASE to increase the salary by 10% for employees who have a hire\_date earlier or equal to 1990.

# Appendix

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated