

SQL Assessment Izabella Paschaly

TASKS

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

My first step was to log onto the database and check the tables have all the data in them and where all the data is located.

CODE

```
USE database employees;
SHOW tables
SELECT * FROM departments;
SELECT * FROM dept_emp;
SELECT * FROM dept_manager;
SELECT * FROM employees;
SELECT * FROM salaries;
```

After the tables were viewed, the LEFT JOIN was used, all managers wanted to be listed, this included the NULL results even if the title or name was not included in the table.

```
SELECT CONCAT(e.first_name, ' ', e.last_name) AS full_name, t.title, d.
emp_no
FROM
dept_manager d
LEFT JOIN
employees e
ON d.emp_no = e.emp_no
LEFT JOIN
titles t
ON t.emp_no = d.emp_no;
```

RESULT OF THE CODE

full_name	title	emp_no
Georgi Facello	Senior Engineer	10001
Bezalel Simmel	Staff	10002
Parto Bamford	Senior Engineer	10003
Saniya Kalloufi	Assistant Engineer	10008
Mary Sluis	NULL	10011
Patricio Bridgland	NULL	10012
Eberhardt Terkki	NULL	10013
Berni Genin	NULL	10014

8 rows in set (0.003 sec)

An addition step, was to use an INNER JOIN on three tables to link managers to their name and their titles, the INNER JOIN eliminated all NULL results.

CODE

```

SELECT d.emp_no, CONCAT(e.first_name, ' ', e.last_name) AS full_name, t.
title
FROM
dept_manager d
INNER JOIN
employees e
ON d.emp_no = e.emp_no
INNER JOIN
titles t
ON t.emp_no = d.emp_no;

```

RESULT OF THE CODE:

emp_no	full_name	title
10001	Georgi Facello	Senior Engineer
10002	Bezalel Simmel	Staff
10003	Parto Bamford	Senior Engineer
10008	Saniya Kalloufi	Assistant Engineer

4 rows in set (0.001 sec)

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

For this question, tables employees, salaries and department names need to link together, as the department tables have department numbers and no employee IDs.

First INNER JOIN linked the employees to their salary

Second INNER JOIN linked the employees' ID to their departments

Third INNER JOIN linked the department numbers to their department names.

CODE

```

SELECT CONCAT(e.first_name, ' ', e.last_name) AS full_name, s.salary, g.
dept_name
FROM employees e
INNER JOIN
salaries s
ON e.emp_no = s.emp_no
INNER JOIN
dept_emp f
ON e.emp_no = f.emp_no
INNER JOIN
departments g
ON f.dept_no = g.dept_no;

```

RESULT OF THE CODE:

full_name	salary	dept_name
Georgi Facello	60117	Development
Georgi Facello	62102	Development
Bezalel Simmel	66074	Sales
Parto Bamford	66596	Production
Chirstian Koblick	66961	Production
Kyoichi Maliniak	71046	Human Resources
Anneke Preusig	74333	Development

7 rows in set (0.001 sec)

However, the results above do not show all the employees. The below code was used to list all employees, including NULL results. The LEFT JOIN linked all the employees from the employees' table and matched these to the other tables.

CODE

```
SELECT CONCAT(e.first_name, ' ', e.last_name) AS full_name, s.salary, g.
dept_name
FROM employees e
LEFT JOIN
salaries s
ON e.emp_no = s.emp_no
LEFT JOIN
dept_emp f
ON e.emp_no = f.emp_no
LEFT JOIN
departments g
ON f.dept_no =g.dept_no;
```

RESULT OF CODE

full_name	salary	dept_name
Georgi Facello	60117	Development
Georgi Facello	62102	Development
Bezalel Simmel	66074	Sales
Parto Bamford	66596	Production
Chirstian Koblick	66961	Production
Kyoichi Maliniak	71046	Human Resources
Anneke Preusig	74333	Development
Tzvetan Zielinski	75286	NULL
Saniya Kalloufi	75994	NULL
Sumant Peac	NULL	NULL
Duangkaew Piveteau	NULL	NULL
Mary Sluis	NULL	NULL
Patricio Bridgland	NULL	NULL
Eberhardt Terkki	NULL	NULL
Berni Genin	NULL	Development

The result above shows Georgi Facello with two salaries, this was due to Georgi having two salaries in the salary table. To remove the duplicate a MAX(salary) and GROUP BY was used, this removed the old salary. If an employee had a pay decrease as their latest salary, the MAX() will need to link to the from-date column in the salary, as this will give the latest salary not their maximum salary.

CODE

```

SELECT CONCAT(e.first_name, ' ', e.last_name) AS full_name,
MAX(s.salary) AS salary, g.dept_name
FROM
employees e
LEFT JOIN
salaries s
ON e.emp_no = s.emp_no
LEFT JOIN
dept_emp f
ON e.emp_no = f.emp_no
LEFT JOIN
departments g
ON f.dept_no = g.dept_no
GROUP BY full_name, g.dept_name;

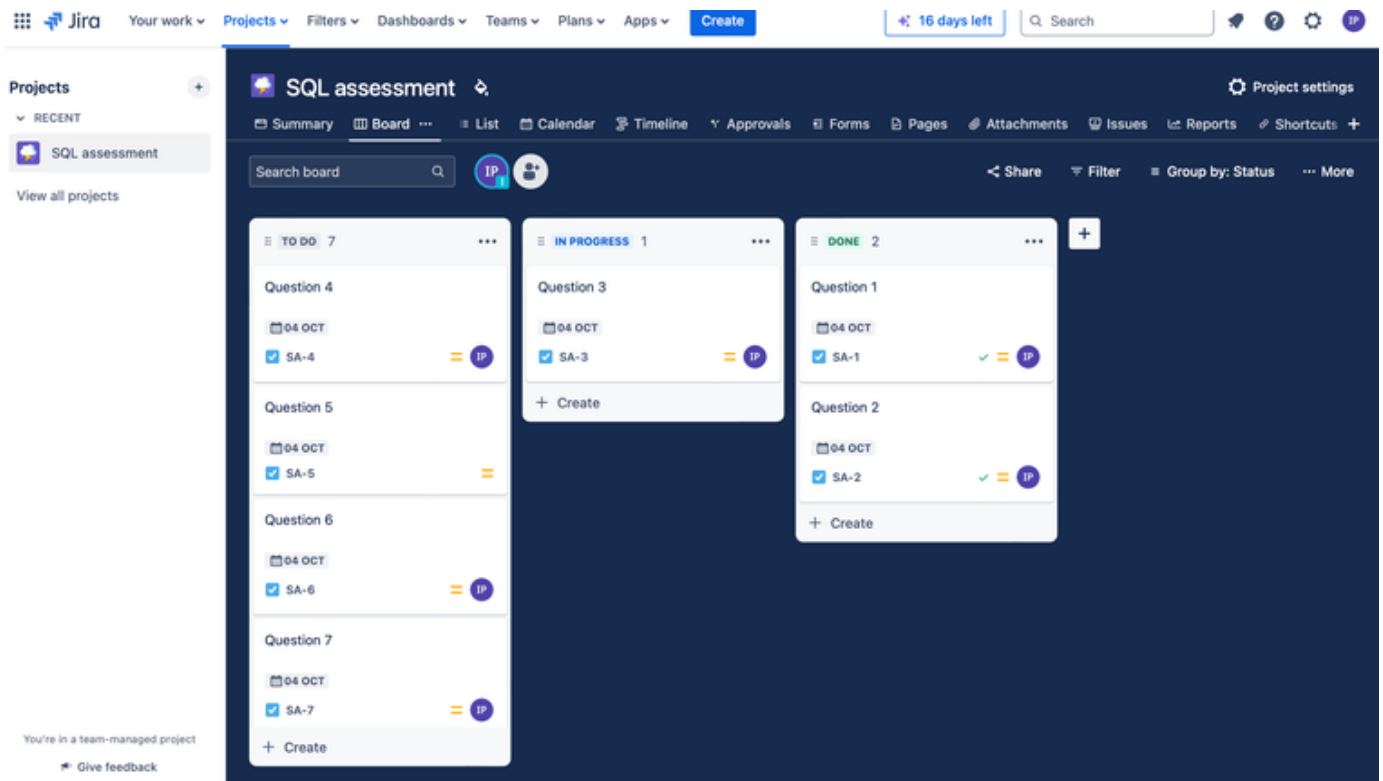
```

RESULT OF CODE (duplicate removed)

full_name	salary	dept_name
Anneke Preusig	74333	Development
Berni Genin	NULL	Development
Bezalel Simmel	66074	Sales
Chirstian Koblick	66961	Production
Duangkaew Piveteau	NULL	NULL
Eberhardt Terkki	NULL	NULL
Georgi Facello	62102	Development
Kyoichi Maliniak	71046	Human Resources
Mary Sluis	NULL	NULL
Parto Bamford	66596	Production
Patricio Bridgland	NULL	NULL
Saniya Kalloufi	75994	NULL
Sumant Peac	NULL	NULL
Tzvetan Zielinski	75286	NULL

14 rows in set (0.003 sec)

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



The above image shows my progress being recorded in JIRA.

All information is spread between these tables:

Hire date - table employees,

birth date - table employees,

HR depart - table departments, but this must link to table dept_emp as this table has the employees' numbers in it, which is also included in the employees' table. INNER JOIN will link all the tables together. The UNION function linked the two separate INNER JOIN and WHERE queries to one table. The CONCAT was used to merge the employees first and last name.

CODE

```
SELECT CONCAT(e.first_name, ' ', e.last_name) AS full_name, e.
birth_date, e.hire_date, g.dept_name
FROM employees e
INNER JOIN dept_emp f ON e.emp_no = f.emp_no
INNER JOIN departments g ON f.dept_no = g.dept_no
WHERE g.dept_name = 'Human Resources'

UNION

SELECT CONCAT(e.first_name, ' ', e.last_name) AS full_name, e.
birth_date, e.hire_date, g.dept_name
FROM employees e
INNER JOIN dept_manager d ON e.emp_no = d.emp_no
INNER JOIN departments g ON d.dept_no = g.dept_no
WHERE g.dept_name = 'Human Resources';
```

RESULT OF THE CODE

full_name	birth_date	hire_date	dept_name
Kyoichi Maliniak	1955-01-21	1989-09-12	Human Resources
Mary Sluis	1953-11-07	1990-01-22	Human Resources
Patricio Bridgland	1960-10-04	1992-12-18	Human Resources

3 rows in set (0.027 sec)

1. Create a SQL statement to show all departments and their department's managers.

My first step was to link table departments to dept_manager, as the dept_manager table had both the department number and employee ID (this linked the manager to the department name and number.) INNER JOIN one. Then the next step was to link the department managers to their names, which were found in the table employees. INNER JOIN two.

CODE

```
SELECT CONCAT(e.first_name, ' ', e.last_name) AS full_name, g.dept_name
FROM
employees e
INNER JOIN
dept_manager d
ON e.emp_no = d.emp_no
INNER JOIN
departments g
ON g.dept_no = d.dept_no;
```

RESULT OF THE CODE

full_name	dept_name
Georgi Facello	Marketing
Bezalel Simmel	Finance
Parto Bamford	Production
Saniya Kalloufi	Finance
Mary Sluis	Human Resources
Patricio Bridgland	Human Resources
Eberhardt Terkki	Marketing
Berni Genin	Production

8 rows in set (0.003 sec)

The below code was used to get all departments and the manager, including NULL results. The LEFT JOIN linked all the departments and matched the department name and number to the managers. The NULL result showed that some departments had no managers. The RANK and PARTITION split the data, the PARTITION BY split the data into groups (partitions) for calculations. Each department was treated separately, this showed both the old and new managers. The RANK BY was to assign a rank to each row within departments. If two rows have the same value, they get the same rank, and the next rank will skip a number (e.g., 1, 1, 3). The new managers had a rank of 1 and the old managers had a rank of 2.

CODE

```

SELECT g.dept_name, d.from_date, d.to_date,
CONCAT(e.first_name, ' ', e.last_name) AS full_name,
MAX(d.from_date) OVER (PARTITION BY g.dept_name) AS max_from_date,
RANK() OVER (PARTITION BY g.dept_name ORDER BY d.from_date DESC) AS rank
FROM
departments g
LEFT JOIN
dept_manager d ON g.dept_no = d.dept_no
LEFT JOIN
employees e ON d.emp_no = e.emp_no;

```

RESULT OF THE CODE

dept_name	full_name	from_date	to_date	max_from_date	rank
Customer Service	NULL	NULL	NULL	NULL	1
Development	NULL	NULL	NULL	NULL	1
Finance	Saniya Kalloufi	1989-12-17	9999-01-01	1989-12-17	1
Finance	Bezalel Simmel	1985-01-01	1989-12-17	1989-12-17	2
Human Resources	Mary Sluis	1992-03-21	9999-01-01	1992-03-21	1
Human Resources	Patricio Bridgland	1985-01-01	1992-03-21	1992-03-21	2
Marketing	Georgi Facello	1991-10-01	9999-01-01	1991-10-01	1
Marketing	Eberhardt Terkki	1985-01-01	1991-10-01	1991-10-01	2
Production	Parto Bamford	1988-09-09	1992-08-02	1988-09-09	1
Production	Berni Genin	1985-01-01	1988-09-09	1988-09-09	2
Quality Management	NULL	NULL	NULL	NULL	1
Research	NULL	NULL	NULL	NULL	1
Sales	NULL	NULL	NULL	NULL	1

To remove the old managers, a MAX(), was used on the from date, and a GROUP BY was used to eliminate the old managers and only show the most current manager.

CODE

```

SELECT g.dept_name, CONCAT(e.first_name, ' ', e.last_name) AS
full_name,
MAX(d.from_date) AS from_date, d.to_date
FROM departments g
LEFT JOIN
dept_manager d
ON g.dept_no = d.dept_no
LEFT JOIN employees e
ON d.emp_no = e.emp_no
GROUP BY g.dept_name, d.dept_no;

```

RESULT OF THE CODE

dept_name	full_name	from_date	to_date
Customer Service	NULL	NULL	NULL
Development	NULL	NULL	NULL
Finance	Bezalel Simmel	1989-12-17	1989-12-17
Human Resources	Mary Sluis	1992-03-21	9999-01-01
Marketing	Georgi Facello	1991-10-01	9999-01-01
Production	Parto Bamford	1988-09-09	1992-08-02
Quality Management	NULL	NULL	NULL
Research	NULL	NULL	NULL
Sales	NULL	NULL	NULL

9 rows in set (0.009 sec)

The screenshot shows the Jira interface for a project named 'SQL assessment'. The main view is a Kanban board with three columns: 'TO DO' (5 items), 'IN PROGRESS' (1 item), and 'DONE' (4 items). Each item represents a question (Question 6 through Question 9, and Question 1 through Question 4) with a due date of '04 OCT' and a status (SA-6, SA-7, SA-8, SA-9, SA-5, SA-1, SA-2, SA-3, SA-4). The board includes a search bar, filters, and a '16 days left' timer. The left sidebar shows the project list with 'SQL assessment' selected. The top navigation bar includes links to 'Your work', 'Projects', 'Filters', 'Dashboards', 'Teams', 'Plans', 'Apps', and a 'Create' button.

1. Create a SQL statement to show a list of HR employees who were hired after 1986

Information needed for this question came from tables departments (for department names and numbers), dept_emp (this linked the department number and employee ID), dept_managers (this linked the department number and employee ID) and employees (employees' names and hire date).

The first INNER JOIN was to link the employees to the HR department, this linked their employees' names and hire dates to the correct department (this code ran on its own and showed all the departments.)

The second INNER JOIN was to link the managers to the HR department, this linked the manager's names and hire dates to the correct department (this code ran on its own and showed all the departments.)

The UNION function linked the two separate INNER JOIN and WHERE queries to one table, this linked the rows from the two INNER JOINs into one table.

I used two WHERE functions, which specified the department name and the year.

CODE


```

SELECT CONCAT(e.first_name, ' ', e.last_name) AS full_name,
e.hire_date, g.dept_name
FROM employees e
INNER JOIN dept_emp f ON e.emp_no = f.emp_no
INNER JOIN departments g ON f.dept_no = g.dept_no
WHERE g.dept_name = 'Human Resources'

UNION

SELECT CONCAT(e.first_name, ' ', e.last_name) AS full_name, e.
hire_date, g.dept_name
FROM employees e
INNER JOIN dept_manager d ON e.emp_no = d.emp_no
INNER JOIN departments g ON d.dept_no = g.dept_no
WHERE g.dept_name = 'Human Resources' AND e.hire_date > '1986-12-12';

```

RESULT OF THE CODE

full_name	hire_date	dept_name
Kyoichi Maliniak	1989-09-12	Human Resources
Mary Sluis	1990-01-22	Human Resources
Patricio Bridgland	1992-12-18	Human Resources

3 rows in set (0.001 sec)

1. 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.

The tables needed for this question are employees and salaries. The UPDATE changed the salary column by a two percent increase, the RIGHT JOIN linked the employee number on the salaries table to the employee's last name, which included NULL results. The WHERE will filter the SELECT function to only a certain last name.

The below code showed the temporary table created, as I did not want to update the original table. The table included data from the salaries and employees tables. The SELECT * showed the temporary table information. The GROUP BY was used with the MAX(), as this showed the maximum salary for each combination of last name and employee number.

CODE

```

DROP TEMPORARY TABLE IF EXISTS temp_salaries;
CREATE TEMPORARY TABLE temp_salaries AS
SELECT e.first_name, e.last_name, s.emp_no, MAX(s.salary) as salary
FROM salaries s
RIGHT JOIN employees e
ON s.emp_no = e.emp_no
GROUP BY e.last_name, s.emp_no;

SELECT *
FROM temp_salaries;

```

RESULT OF THE CODE

first_name	last_name	emp_no	salary
Parto	Bamford	10003	66596
Patricio	Bridgland	NULL	NULL
Georgi	Facello	10001	62102
Berni	Genin	NULL	NULL
Saniya	Kalloufi	10008	75994
Chirstian	Koblick	10004	66961
Kyoichi	Maliniak	10005	71046
Sumant	Peac	NULL	NULL
Duangkaew	Piveteau	NULL	NULL
Anneke	Preusig	10006	74333
Bezalel	Simmel	10002	66074
Mary	Sluis	NULL	NULL
Eberhardt	Terkki	NULL	NULL
Tzvetan	Zielinski	10007	75286

14 rows in set (0.000 sec)

After the temporary table was created, the UPDATE function changed the salary of Bamford on the temporary table but did not affect the original data

CODE

```
UPDATE temp_salaries ts
SET ts.salary = ts.salary * 1.02
WHERE ts.last_name = 'Bamford';

SELECT *
FROM temp_salaries;
```

The updated salary for the employee with the last name Bamford, this field can be changed to any other employee's last name. The salary increased from 66596 to 67928. This can be seen in the table below.

RESULT OF THE CODE for the temporary table

first_name	last_name	emp_no	salary
Parto	Bamford	10003	67928
Patricio	Bridgland	NULL	NULL
Georgi	Facello	10001	62102
Berni	Genin	NULL	NULL
Saniya	Kalloufi	10008	75994
Chirstian	Koblick	10004	66961
Kyoichi	Maliniak	10005	71046
Sumant	Peac	NULL	NULL
Duangkaew	Piveteau	NULL	NULL
Anneke	Preusig	10006	74333
Bezalel	Simmel	10002	66074
Mary	Sluis	NULL	NULL
Eberhardt	Terkki	NULL	NULL
Tzvetan	Zielinski	10007	75286

14 rows in set (0.000 sec)

The below table was the original data from the original table, this image showed that updating the temporary table does not affect the original data.

first_name	last_name	emp_no	salary
Parto	Bamford	10003	66596
Patricio	Bridgland	NULL	NULL
Georgi	Facello	10001	62102
Berni	Genin	NULL	NULL
Saniya	Kalloufi	10008	75994
Chirstian	Koblick	10004	66961
Kyoichi	Maliniak	10005	71046
Sumant	Peac	NULL	NULL
Duangkaew	Piveteau	NULL	NULL
Anneke	Preusig	10006	74333
Bezalel	Simmel	10002	66074
Mary	Sluis	NULL	NULL
Eberhardt	Terkki	NULL	NULL
Tzvetan	Zielinski	10007	75286

14 rows in set (0.003 sec)

Jira Your work Projects Filters Dashboards Teams Plans Apps Create 15 days left Search

Projects

RECENT

SQL assessment

View all projects

You're in a team-managed project Give feedback

SQL assessment Project settings

Summary Board List Calendar Timeline Approvals Forms Pages Attachments Issues Reports Shortcuts

Search board IP-1

Share 1 filter applied Group by: Status More

TO DO 3

Question 8

04 OCT

SA-8 = IP

Question 9

04 OCT

SA-9 = IP

Question 10

04 OCT

SA-10 = IP

+ Create

IN PROGRESS 1

Question 7

04 OCT

SA-7 = IP

+ Create

DONE 6

Question 1

04 OCT

SA-1 ✓ = IP

Question 2

04 OCT

SA-2 ✓ = IP

Question 3

04 OCT

SA-3 ✓ = IP

Question 4

04 OCT

SA-4 ✓ = IP

+ Create

Alternate way of doing question six:

The method used for this question involved a CTE (common table expression). This was used instead of creating a temporary table, this could be used for questions seven and ten. An UPDATE and SET cannot be used in a CTE since they are not permanent structures, you cannot modify them. A CASE was used to choose the conditions of the last name.

CODE

```

WITH employee_CTE AS (
SELECT e.first_name, e.last_name, e.emp_no, s.salary,
MAX(s.from_date) AS from_date
FROM
salaries s
RIGHT JOIN
employees e ON s.emp_no = e.emp_no
GROUP BY e.emp_no, e.first_name, e.last_name, s.salary)

SELECT first_name, last_name, emp_no,
CASE
WHEN last_name = 'Bamford' THEN salary * 1.02
ELSE salary
END AS updated_salary, from_date
FROM employee_CTE;

```

RESULT OF CODE

first_name	last_name	emp_no	updated_salary	from_date
Georgi	Facello	10001	60117.00	1986-06-26
Georgi	Facello	10001	62102.00	1987-06-26
Bezalel	Simmel	10002	66074.00	1988-06-25
Parto	Bamford	10003	67927.92	1989-06-25
Chirstian	Koblick	10004	66961.00	1990-06-25
Kyoichi	Maliniak	10005	71046.00	1991-06-25
Anneke	Preusig	10006	74333.00	1992-06-24
Tzvetan	Zielinski	10007	75286.00	1993-06-24
Saniya	Kalloufi	10008	75994.00	1994-06-24
Sumant	Peac	10009	NULL	NULL
Duangkaew	Piveteau	10010	NULL	NULL
Mary	Sluis	10011	NULL	NULL
Patricio	Bridgland	10012	NULL	NULL
Eberhardt	Terkki	10013	NULL	NULL
Berni	Genin	10014	NULL	NULL

15 rows in set (0.005 sec)

1. Create an SQL statement to delete an employee's record who belongs to the marketing department and name start with A

The DELETE function was used to delete the employee date, this was also used with the WHERE function to delete only the information we wanted. The LEFT JOIN was used to find all the employees in the employees table with a name that started with A, then linked the employee to their department. For the WHERE function, A% is a wildcard character which was used to substitute one or more characters in a string. (The % represents zero or more characters.)

The below code was used to create a temporary table, which included the employee number, the employee's name and their department. The LEFT JOIN required two labels (g and gm) since the same table departments are being joined twice (once for employees and once for managers), using different labels is needed. If the same label was used for both joins, SQL wouldn't know which one you were referring to and this would of led to an error.

CODE

```

DROP TEMPORARY TABLE IF EXISTS temp_marketing_employees;
CREATE TEMPORARY TABLE temp_marketing_employees AS
SELECT e.emp_no, e.first_name, e.last_name,
g.dept_name AS employee_dept_name,
gm.dept_name AS manager_dept_name
FROM employees e
LEFT JOIN
dept_emp f
ON e.emp_no = f.emp_no
LEFT JOIN
departments g
ON f.dept_no = g.dept_no
LEFT JOIN
dept_manager d
ON e.emp_no = d.emp_no
LEFT JOIN
departments gm
ON d.dept_no = gm.dept_no;

SELECT *
FROM temp_marketing_employees;

```

RESULT OF THE CODE

emp_no	first_name	last_name	employee_dept_name	manager_dept_name
10001	Georgi	Facello	Development	Marketing
10002	Bezalel	Simmel	Sales	Finance
10003	Parto	Bamford	Production	Production
10004	Chirstian	Koblick	Production	NULL
10005	Kyoichi	Maliniak	Human Resources	NULL
10006	Anneke	Preusig	Development	NULL
10007	Tzvetan	Zielinski	NULL	NULL
10008	Saniya	Kalloufi	NULL	Finance
10009	Sumant	Peac	NULL	NULL
10010	Duangkaew	Piveteau	NULL	NULL
10011	Mary	Sluis	NULL	Human Resources
10012	Patricio	Bridgland	NULL	Human Resources
10013	Eberhardt	Terkki	NULL	Marketing
10014	Berni	Genin	Development	Production
14 rows in set (0.000 sec)				

The below code was to delete the employee whose name started with A and who was in the marketing department, this was deleted from the temporary table, however, no employee had this criteria so the table looks the same.

CODE

```

DELETE
FROM temp_marketing_employees
WHERE first_name LIKE 'A%'
AND (employee_dept_name = 'Marketing' OR manager_dept_name =
'Marketing');

SELECT *
FROM temp_marketing_employees;

```

RESULT OF THE CODE

emp_no	first_name	last_name	employee_dept_name	manager_dept_name
10001	Georgi	Facello	Development	Marketing
10002	Bezalel	Simmel	Sales	Finance
10003	Parto	Bamford	Production	Production
10004	Chirstian	Koblick	Production	NULL
10005	Kyoichi	Maliniak	Human Resources	NULL
10006	Anneke	Preusig	Development	NULL
10007	Tzvetan	Zielinski	NULL	NULL
10008	Saniya	Kalloufi	NULL	Finance
10009	Sumant	Peac	NULL	NULL
10010	Duangkaew	Piveteau	NULL	NULL
10011	Mary	Sluis	NULL	Human Resources
10012	Patricio	Bridgland	NULL	Human Resources
10013	Eberhardt	Terkki	NULL	Marketing
10014	Berni	Genin	Development	Production

14 rows in set (0.000 sec)

1. Create a database view to list the full names of all departments' managers, and their salaries.

Projects

RECENT

SQL assessment

View all projects

SQL assessment

Summary Board List Calendar Timeline Approvals Forms Pages Attachments Issues Reports Shortcuts

Search board

Share 1 filter applied Group by: Status More

TO DO 2

Question 9

04 OCT

SA-9

Question 10

04 OCT

SA-10

Create

IN PROGRESS 1

Question 8

04 OCT

SA-8

Create

DONE 7

Question 1

04 OCT

SA-1

Question 2

04 OCT

SA-2

Question 3

04 OCT

SA-3

Question 4

04 OCT

SA-4

Create

You're in a team-managed project

Give feedback

The DROP VIEW dropped a view if it had the same name, and then a new database view was created. The database view was created by selecting specific columns from the department managers, employees and salaries table. The SELECT * FROM view_name, displayed the results of the view, this included the columns first name, last name, department name and their salary. The CONCAT combined the first and last name of all employees into one column, the ' ', was to leave a space between the first and last name. The MAX function with the GROUP BY, eliminated any duplicates in the salary as the latest salary was chosen.

CODE

```
DROP VIEW IF EXISTS view_deptm_salary;
CREATE VIEW view_deptm_salary AS
SELECT CONCAT (e.first_name, ' ', e.last_name) AS full_name,
g.dept_name, s.salary, MAX(s.from_date) as from_date
FROM dept_manager d
LEFT JOIN employees e
ON d.emp_no = e.emp_no
LEFT JOIN departments g
ON d.dept_no = g.dept_no
LEFT JOIN salaries s
ON d.emp_no = s.emp_no
GROUP BY full_name, g.dept_name;

SELECT *
FROM view_deptm_salary;
```

RESULT OF THE CODE

full_name	dept_name	salary	from_date
Berni Genin	Production	NULL	NULL
Bezalel Simmel	Finance	66074	1988-06-25
Eberhardt Terkki	Marketing	NULL	NULL
Georgi Facello	Marketing	60117	1987-06-26
Mary Sluis	Human Resources	NULL	NULL
Parto Bamford	Production	66596	1989-06-25
Patricio Bridgland	Human Resources	NULL	NULL
Saniya Kalloufi	Finance	75994	1994-06-24

8 rows in set (0.004 sec)

To remove the NULL values and leave only the current managers, instead of a LEFT JOIN replace LEFT JOIN with INNER JOIN. The results will be four department managers.

1. Create a database view to list all departments and their department's managers, who were hired between 1980 and 1990.

The DROP VIEW dropped a view if it had the same name, and then a new database view was created. The LEFT JOINs linked all departments to the correct manager (this result included NULL values). The WHERE clause showed the managers hired between 01-01-1980 and 31-12-1990. The ORDER BY clause ordered the departments so the departments were listed together, as well listed the dates the managers were hired in ascending order.

CODE


```

DROP VIEW IF EXISTS view_managers_hiredate;
CREATE VIEW view_managers_hiredate AS
SELECT g.dept_name, CONCAT (e.first_name, ' ', e.last_name) AS
full_name, d.from_date, d.to_date
FROM departments g
LEFT JOIN
dept_manager d
ON g.dept_no = d.dept_no
LEFT JOIN employees e
ON d.emp_no = e.emp_no
WHERE e.hire_date BETWEEN '1980-01-01' AND '1990-12-21'
ORDER BY g.dept_name, d.from_date;

SELECT *
FROM view_managers_hiredate;

```

RESULT OF THE CODE

dept_name	full_name	from_date	to_date
Finance	Bezalel Simmel	1985-01-01	1989-12-17
Human Resources	Mary Sluis	1992-03-21	9999-01-01
Marketing	Eberhardt Terkki	1985-01-01	1991-10-01
Marketing	Georgi Facello	1991-10-01	9999-01-01
Production	Berni Genin	1985-01-01	1988-09-09
Production	Parto Bamford	1988-09-09	1992-08-02

6 rows in set (0.004 sec)

1. Create a SQL statement to increase the salaries of all department managers up to 10% who have been working since 1990.

The first part of the code was to create a temporary table with all department managers, salaries and their to date (the to date is needed to see if they left before 1990.)

CODE


```

DROP TEMPORARY TABLE IF EXISTS temp_manager_increase;

CREATE TEMPORARY TABLE temp_manager_increase AS
SELECT CONCAT (e.first_name, ' ', e.last_name) AS full_name,
MAX(s.salary) AS salary, d.from_date
FROM
dept_manager d
LEFT JOIN
employees e
ON d.emp_no = e.emp_no
LEFT JOIN
salaries s
ON s.emp_no = e.emp_no
GROUP BY full_name;

SELECT *
FROM temp_manager_increase;

```

RESULT OF THE CODE

full_name	salary	from_date
Berni Genin	NULL	1985-01-01
Bezalel Simmel	66074	1985-01-01
Eberhardt Terkki	NULL	1985-01-01
Georgi Facello	62102	1991-10-01
Mary Sluis	NULL	1992-03-21
Parto Bamford	66596	1988-09-09
Patricio Bridgland	NULL	1985-01-01
Saniya Kalloufi	75994	1989-12-17

8 rows in set (0.000 sec)

The code below was to update the salaries of all department managers who worked after 1989-12-31, this date was chosen for the WHERE function, so 1990-01-01 was included. Alternatively, a >= could have been used, with the date 1990-01-01. The duplicated salary of Georgi Facello was removed as only his most current salary was updated, as he had two salaries after 1990. Georgi was the only manager since 1990, his salary increase from 62102 to 68312 included as he had two salary increases since 1990, so both needed

CODE

```

UPDATE temp_manager_increase t
SET t.salary = t.salary * 1.10
WHERE from_date > '1989-12-31';

SELECT *
FROM temp_manager_increase;

```

RESULT OF CODE

full_name	salary	from_date
Berni Genin	NULL	1985-01-01
Bezalel Simmel	66074	1985-01-01
Eberhardt Terkki	NULL	1985-01-01
Georgi Facello	68312	1991-10-01
Mary Sluis	NULL	1992-03-21
Parto Bamford	66596	1988-09-09
Patricio Bridgland	NULL	1985-01-01
Saniya Kalloufi	75994	1989-12-17

8 rows in set (0.000 sec)

Projects

RECENT

SQL assessment

View all projects

SQL assessment

Project settings

SummaryBoardListCalendarTimelineApprovalsFormsPagesAttachmentsIssuesReportsShortcuts

Search boardIP

Share1 filter appliedGroup by: StatusMore

TO DO 0

IN PROGRESS 0

DONE 10

Question 1

04 OCT

SA-1

Question 2

04 OCT

SA-2

Question 3

04 OCT

SA-3

Question 4

04 OCT

SA-4

Create

Create

Create

You're in a team-managed project

Give feedback