

These are some real-life challenges that can be solved using SQL. Some examples at some common issues utilizing Microsoft SQL Suite and Software Salaries dataset are provided in the following PDF

1) Buscar el salario máximo, o el Nth salario que deseemos

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'AGUSBE-PC\SQLEXPRESS01 (SQL Server 16.0.1000 - AC)'. The 'Salarios' database is expanded, showing the 'dbo.salarydata' table. The SQL Query window on the right contains the following T-SQL code:

```
--Buscamos el salario mas alto
SELECT max (salary)
FROM salarydata

--Buscamos el segundo salario mas alto
SELECT salary
FROM salarydata
WHERE salary < (SELECT MAX(salary) FROM salarydata)
ORDER BY salary DESC
OFFSET 1 ROWS FETCH NEXT 1 ROWS ONLY;
```

The Results pane shows the output of the queries. The first query returns a single row with the maximum salary: 90000000. The second query returns a single row with the second highest salary: 10000000.

2) Elaborar un ranking de salarios de los distintos empleados

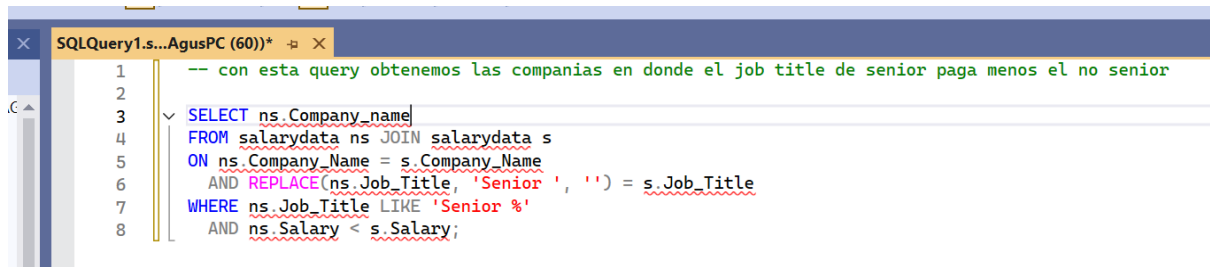
The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for 'AGUSBE-PC\SQLEXPRESS01 (SQL Server 16.0.1000 - AC)'. The 'Salarios' database is expanded, showing the 'dbo.salarydata' table. The SQL Query window on the right contains the following T-SQL code:

```
-- Rankeamos salarios de mayor a menor, con rango compartido para iguales valores
SELECT *, DENSE_RANK() OVER (ORDER BY salary DESC) AS Rank
FROM salarydata;
```

The Results pane shows the output of the query, displaying a list of employees with their salaries and ranks. The results are sorted by salary in descending order. The first row shows an employee with a salary of 90000000 and a rank of 1. The last row shows an employee with a salary of 8700000 and a rank of 9.

Rating	Company_Name	Job_Title	Salary	Salaries_Reported	Location	Employment_Status	Job_Roles	Rank	
1	36	Thapar University	Software Development Engineer (SDE)	90000000	1	New Delhi	Full Time	SDE	1
2	38	Concentrix	Oracle Database Administrator	10000000	1	Bangalore	Full Time	Database	2
3	35	Koru UX Design	Senior Front End Developer	10000000	1	Pune	Full Time	Frontend	2
4	36	OASYS Cybernetics	Senior Java Developer	10000000	1	Chennai	Full Time	Java	2
5	37	Niyo Infotech	Lead UI Designer, Magento Front-end Developer	9900000	1	Bangalore	Full Time	Frontend	3
6	38	Amazon	Software Development Engineer (SDE)	9850000	1	Kolkata	Full Time	SDE	4
7	42	FFF Enterprises	Non Software Development Engineer	9800000	1	Mumbai	Full Time	SDE	5
8	43	Digital Raju	Software Development Engineer (SDE)	9800000	1	New Delhi	Full Time	SDE	5
9	50	Gaana Lyrics Point.com	Software Development Engineer (SDE) II	9700000	1	Hyderabad	Full Time	SDE	6
10	46	GGH Heavy Industries	Best Buy Mobile Sales Associate	9600000	1	Bangalore	Full Time	Mobile	7
11	39	NortonLifeLock	Software Development Engineer In Test (SDET)	9400000	1	Pune	Full Time	SDE	8
12	35	ArisGlobal	Software Development Engineer	8700000	1	Bangalore	Full Time	SDE	9

- 3) Encontrar las empresas en donde el title de senior paga menos que el de no senior usando SelfJoin

A screenshot of a SQL query editor window titled 'SQLQuery1.s...AgusPC (60))'. The query is written in a monospaced font with syntax highlighting. It starts with a comment on line 1: '-- con esta query obtenemos las companias en donde el job title de senior paga menos el no senior'. The query itself is a self-join on a table named 'salarydata'. It selects the company name from the 'ns' alias. The join is performed on the company name. The where clause filters for senior job titles (using REPLACE and LIKE) and ensures the salary of the senior employee is less than the salary of the non-senior employee.

```
1  -- con esta query obtenemos las companias en donde el job title de senior paga menos el no senior
2
3  SELECT ns.Company_name
4  FROM salarydata ns JOIN salarydata s
5  ON ns.Company_Name = s.Company_Name
6  AND REPLACE(ns.Job_Title, 'Senior ', '') = s.Job_Title
7  WHERE ns.Job_Title LIKE 'Senior %'
8  AND ns.Salary < s.Salary;
```

Basado en el ejemplo de una tabla en donde comparamos salarios del manager con sus empleados, hay dos opciones para realizar la misma query

SELECT E.NAME

FROM EMPLOYEE E

JOIN EMPLOYEE M ON E.MANAGERID = M.ID

WHERE E.SALARY > M.SALARY;

O usando CTEs

WITH SALARY_EMPLOYEE AS (

SELECT ID, NAME, SALARY, MANAGERID

FROM EMPLOYEE),

SALARY_MANAGER AS (

SELECT ID, SALARY

FROM EMPLOYEE)

SELECT E.NAME, E.SALARY, M.SALARY AS MANAGER_SALARY

FROM SALARY_EMPLOYEE E JOIN SALARY_MANAGER M ON E.MANAGERID = M.ID

WHERE E.SALARY > M.SALARY;

4) HACER UN LISTADO DE COMPAÑÍAS EN DONDE NO EXISTAN DUPLICADOS

The screenshot shows the SQL Server Enterprise Manager interface. The Object Explorer on the left displays the database structure for AGUSBE-PC\SQLEXPRESS01. The main window shows a query in the SQLQuery1.s...AgusPC (57) tab:

```
1 SELECT Company_Name
2 FROM Salaries
3 WHERE Company_Name IN (
4     SELECT Company_Name
5     FROM Salaries
6     GROUP BY Company_Name
7     HAVING COUNT(*) > 1
8 );
9
```

The Results pane at the bottom shows the output of the query, listing 12 companies:

Company_Name
1 Sasken
2 Unacademy
3 Appoids Tech Solutions
4 Freelancer
5 Samsung R&D Institute India - Bangalore
6 DXMinds Technologies
7 Craft Silicon
8 Baronford & Associates
9 Wibmo
10 Bookmyshow
11 Knowledge Flex
12 Novocav Solutions

The status bar at the bottom indicates the query was executed successfully.

5) Salario mas alto por locación

The screenshot shows the SQL Server Enterprise Manager interface. The main window shows a query in the SQLQuery1.s...AgusPC (57) tab:

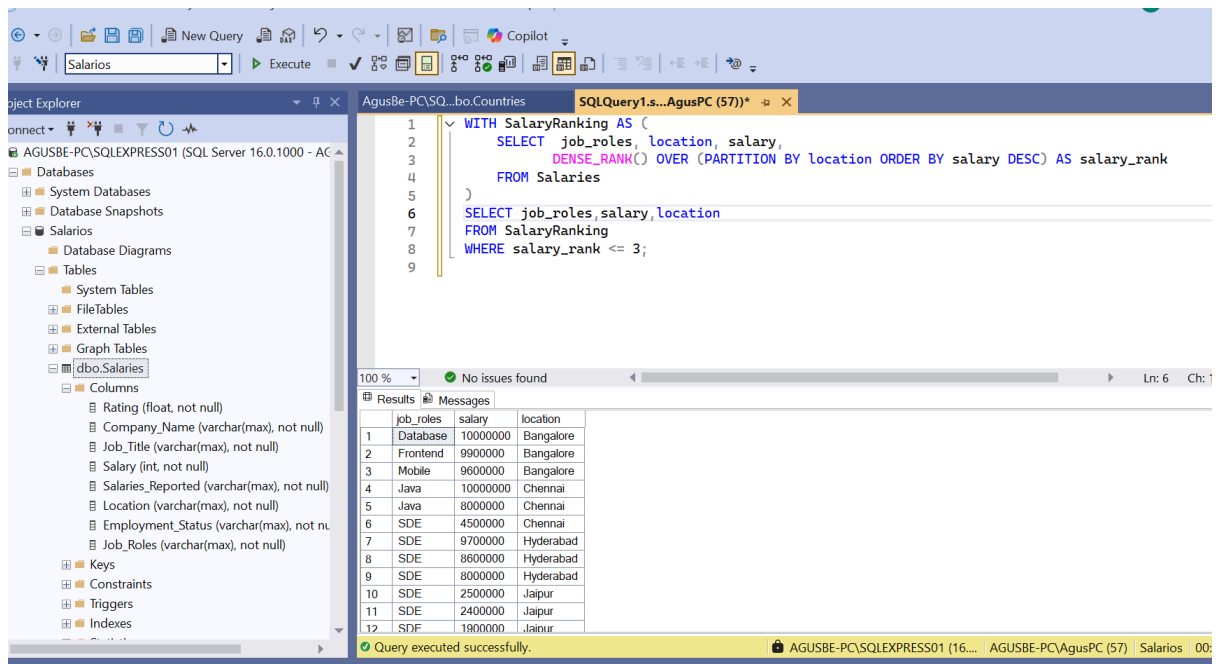
```
1 SELECT location, MAX(salary) AS highest_salary
2 FROM Salaries
3 GROUP BY location;
4
```

The Results pane at the bottom shows the output of the query, listing 10 locations and their highest salaries:

location	highest_salary
1 New Delhi	90000000
2 Pune	10000000
3 Hyderabad	9700000
4 Kerala	2900000
5 Bangalore	10000000
6 Mumbai	9800000
7 Madhya Pradesh	7800000
8 Chennai	10000000
9 Kolkata	9850000
10 Jaipur	2500000

The status bar at the bottom indicates no issues were found.

6) Los 3 salarios mas altos por locación y su respectivo rol de trabajo



The screenshot shows the SQL Server Enterprise Manager interface. On the left, the 'Object Explorer' pane displays the database structure for 'AGUSBE-PC\SQLEXPRESS01 (SQL Server 16.0.1000 - AC)'. The 'Salaries' table is selected under 'dbo'. The main pane shows a SQL query in the 'SQLQuery1.s...AgusPC (57))' window. The query is as follows:

```
1 WITH SalaryRanking AS (  
2     SELECT job_roles, location, salary,  
3           DENSE_RANK() OVER (PARTITION BY location ORDER BY salary DESC) AS salary_rank  
4     FROM Salaries  
5 )  
6 SELECT job_roles, salary, location  
7 FROM SalaryRanking  
8 WHERE salary_rank <= 3;  
9
```

Below the query, the 'Results' pane displays the output of the query. The results are as follows:

	job_roles	salary	location
1	Database	10000000	Bangalore
2	Frontend	9900000	Bangalore
3	Mobile	9800000	Bangalore
4	Java	10000000	Chennai
5	Java	8000000	Chennai
6	SDE	4500000	Chennai
7	SDE	9700000	Hyderabad
8	SDE	8600000	Hyderabad
9	SDE	8000000	Hyderabad
10	SDE	2500000	Jaipur
11	SDE	2400000	Jaipur
12	SDE	1900000	Jaipur

The status bar at the bottom indicates 'Query executed successfully.' and shows the connection to 'AGUSBE-PC\SQLEXPRESS01 (16...' and the database 'AGUSBE-PC\AgusPC (57) - Salaries'.

7) Eliminar filas en donde las posiciones solo sean Intern

DELETE FROM Salaries

WHERE Employment_status = Intern