



SCIENCEQTECH

EMPLOYEE PERFORMANCE MAPPING

SQL PROJECT: END-TO-END
DATA EXPLORATION AND
OPTIMIZATION



ABOUT SCIENCEQTECH!

ScienceQtech is a startup that works in the Data Science field. ScienceQtech has worked on fraud detection, market basket, self-driving cars, supply chain, algorithmic early detection of lung cancer, customer sentiment, and the drug discovery field. With the annual appraisal cycle around the corner, the HR department has asked you (Junior Database Administrator) to generate reports on employee details, their performance, and on the project that the employees have undertaken, to analyze the employee database and extract specific data based on different requirements



PURPOSE & RESPONSIBILITIES

Primary Objective

To extract, analyze, and optimize employee and project-related data using SQL Server, with a focus on enhancing data accessibility, integrity, and performance for informed business decision-making.

Role: Data Analyst

As a Data Analyst, my role is to transform raw data into actionable insights through structured queries, data modeling, and optimization techniques. I ensure data accuracy and build efficient query systems for performance-driven analysis.

Tasks Included

- Wrote and optimized SQL queries to extract and analyze employee and project data.
- Implemented stored procedures, views, functions, and indexing for performance.
- Applied advanced techniques like CTEs and window functions for deeper insights



PURPOSE & RESPONSIBILITIES

Key Deliverables

- Delivered a full set of categorized SQL solutions to 20+ business problems.
- Created reusable database objects like views, functions, and procedures.
- Improved data retrieval speed and structure through indexing and optimization.

Important Points

- Ensured accuracy, consistency, and performance in data querying.
- Built scalable, reusable query logic aligned with real business needs.
- Enabled data-driven decisions through structured, efficient analysis.



DATA PROCESSING

DATASET OVERVIEW

employeeRecord



It contains the information of all the employees.

projectTable



dataScienceTeam

It contains information about all the employees in the Data Science team.



SCIENCEQTECH



FOUNDATIONAL DATA RETRIEVAL AND FILTERING

This section focuses on writing basic to intermediate SQL queries to extract specific employee and project data. It includes operations like filtering records based on conditions, sorting results by specific fields, combining outputs from multiple teams or departments, and summarizing key metrics within each group. These queries lay the groundwork for understanding and exploring raw organizational data.



REQUEST 1

Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, and DEPARTMENT from the employee record table, and make a list of employees and details of their department.

```
SELECT  
EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPT  
FROM employeeRecord
```

	EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT
1	E001	Arthur	Black	M	ALL
2	E005	Eric	Hoffman	M	FINANCE
3	E010	William	Butler	M	AUTOMOTIVE
4	E052	Dianna	Wilson	F	HEALTHCARE
5	E057	Dorothy	Wilson	F	HEALTHCARE
6	E083	Patrick	Voltz	M	HEALTHCARE
7	E103	Emily	Grove	F	FINANCE
8	E204	Karene	Nowak	F	AUTOMOTIVE
9	E245	Nian	Zhen	M	RETAIL
10	E260	Roy	Collins	M	RETAIL



REQUEST 2

Write a query to fetch EMP_ID, FIRST_NAME, LAST_NAME, GENDER, DEPARTMENT, and EMP_RATING if the EMP_RATING is:

- Less than two
- Greater than four
- Between two and four

```
SELECT  
EMP_ID, FIRST_NAME, LAST_NAME,  
GENDER, DEPT, EMP_RATING  
FROM employeeRecord  
WHERE EMP_RATING < 2 OR  
EMP_RATING > 4 OR  
EMP_RATING BETWEEN 2 AND 4  
ORDER BY EMP_RATING ASC
```

	EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT	EMP_RATING
1	E057	Dorothy	Wilson	F	HEALTHCARE	1
2	E532	Claire	Brennan	F	AUTOMOTIVE	1
3	E620	Katrina	Allen	F	RETAIL	1
4	E583	Janet	Hale	F	RETAIL	2
5	E245	Nian	Zhen	M	RETAIL	2
6	E010	William	Butler	M	AUTOMOTIVE	2
7	E505	Chad	Wilson	M	HEALTHCARE	2
8	E005	Eric	Hoffman	M	FINANCE	3
9	E260	Roy	Collins	M	RETAIL	3
10	E403	Steve	Hoffman	M	FINANCE	3



REQUEST 3

Write a query to concatenate the FIRST_NAME and the LAST_NAME of employees in the Finance department from the employee table and then give the resultant column alias as NAME

```
SELECT  
    CONCAT(First_name, ' ', LAST_NAME) AS Name  
FROM employeeRecord  
WHERE DEPT = 'Finance';
```

	Name
1	Eric Hoffman
2	Emily Grove
3	Steve Hoffman



REQUEST 4

Write a query to list down all the employees from the healthcare and finance departments using union take data from the employee record table.

```
SELECT  
    EMP_ID, FIRST_NAME,  
    LAST_NAME, GENDER, DEPT  
FROM employeeRecord  
WHERE DEPT = 'healthcare'  
UNION  
SELECT  
    EMP_ID, FIRST_NAME,  
    LAST_NAME, GENDER, DEPT  
FROM employeeRecord  
WHERE DEPT = 'Finance'
```

	EMP_ID	FIRST_NAME	LAST_NAME	GENDER	DEPT
1	E052	Dianna	Wilson	F	HEALTHCARE
2	E057	Dorothy	Wilson	F	HEALTHCARE
3	E083	Patrick	Voltz	M	HEALTHCARE
4	E505	Chad	Wilson	M	HEALTHCARE
5	E005	Eric	Hoffman	M	FINANCE
6	E103	Emily	Grove	F	FINANCE
7	E403	Steve	Hoffman	M	FINANCE



REQUEST 5

Write a query to list down employee details such as EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPARTMENT, and EMP_RATING grouped by dept. Also, include the respective employee rating along with the max emp rating for the department.

```
SELECT  
    EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPT, EMP_RATING,  
    MAX(EMP_RATING) OVER(PARTITION BY DEPT) AS [Avg Dept Rating]  
FROM employeeRecord
```

	EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EMP_RATING	Avg Dept Rating
1	E001	Arthur	Black	PRESIDENT	ALL	5	5
2	E204	Karene	Nowak	SENIOR DATA SCIENTIST	AUTOMOTIVE	5	5
3	E010	William	Butler	LEAD DATA SCIENTIST	AUTOMOTIVE	2	5
4	E428	Pete	Allen	MANAGER	AUTOMOTIVE	4	5
5	E532	Claire	Brennan	ASSOCIATE DATA SCIENTIST	AUTOMOTIVE	1	5
6	E103	Emily	Grove	MANAGER	FINANCE	4	4
7	E403	Steve	Hoffman	ASSOCIATE DATA SCIENTIST	FINANCE	3	4
8	E005	Eric	Hoffman	LEAD DATA SCIENTIST	FINANCE	3	4





REQUEST 6

Write a query to calculate the minimum and the maximum salary of the employees in each role.
Take data from the employee record table

```
SELECT  
ROLE, MIN(SALARY) AS [Min Salary], MAX(SALARY) AS [Max Salary]  
FROM employeeRecord  
GROUP BY ROLE
```

	ROLE	Min Salary	Max Salary
1	ASSOCIATE DATA SCIENTIST	4000.00	5000.00
2	JUNIOR DATA SCIENTIST	2800.00	3000.00
3	LEAD DATA SCIENTIST	8500.00	9000.00
4	MANAGER	8500.00	11000.00
5	PRESIDENT	16500.00	16500.00
6	SENIOR DATA SCIENTIST	5500.00	7700.00



REQUEST 7

Write a query to create a view that displays employees in various countries whose salary is more than six thousand. Take data from the employee record table

```
SELECT  
    EMP_ID, FIRST_NAME, LAST_NAME, COUNTRY, SALARY  
FROM employeeRecord  
WHERE SALARY > 6000  
ORDER BY SALARY
```



	EMP_ID	FIRST_NAME	LAST_NAME	COUNTRY	SALARY
1	E245	Nian	Zhen	CHINA	6500.00
2	E260	Roy	Collins	INDIA	7000.00
3	E204	Karene	Nowak	GERMANY	7500.00
4	E057	Dorothy	Wilson	USA	7700.00
5	E005	Eric	Hoffman	USA	8500.00
6	E612	Tracy	Nomis	INDIA	8500.00
7	E010	William	Butler	FRANCE	9000.00



REQUEST 8

Write a nested query to find employees with experience of more than ten years. Take data from the employee record table

```
SELECT  
    EMP_ID, FIRST_NAME, LAST_NAME, EXP  
FROM employeeRecord  
WHERE EXP IN (SELECT EXP  
    FROM employeeRecord  
    WHERE EXP > 10)
```



	EMP_ID	FIRST_NAME	LAST_NAME	EXP
1	E001	Arthur	Black	20
2	E005	Eric	Hoffman	11
3	E010	William	Butler	12
4	E083	Patrick	Voltz	15
5	E103	Emily	Grove	14
6	E428	Pete	Allen	14
7	E583	Janet	Hale	14
8	E612	Tracy	Nomis	13



REQUEST 9

Write a query to calculate the bonus for all the employees, based on their ratings and salaries
(Use the formula: 5% of salary * employee rating)

```
SELECT  
    EMP_ID, SALARY,  
    EMP_RATING,  
    (SALARY *5/100)*EMP_RATING AS BONUS  
FROM employeeRecord
```

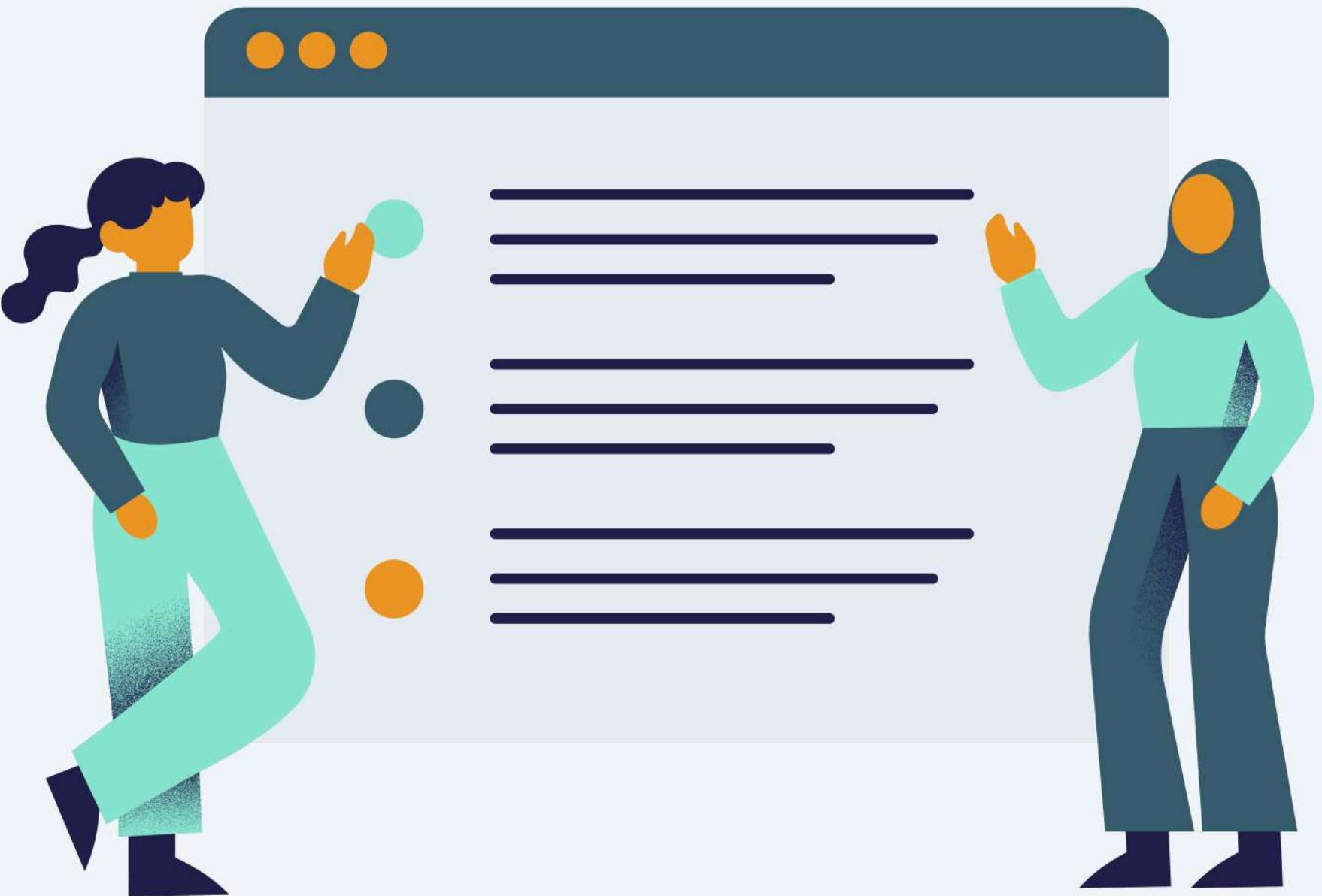


	EMP_ID	SALARY	EMP_RATING	BONUS
1	E001	16500.00	5	4125
2	E005	8500.00	3	1275
3	E010	9000.00	2	900
4	E052	5500.00	5	1375
5	E057	7700.00	1	385
6	E083	9500.00	5	2375



SCHEMA-LEVEL OPERATIONS

Here, the focus shifts from querying to structuring and optimizing the database. Stored functions were created to enforce business logic (like job role mapping), views were built for reusable filtered datasets, and indexes were implemented to boost query performance. This section ensures data logic is embedded at the schema level for efficiency and consistency.



REQUEST 10

Write a query to create a stored procedure to retrieve the details of the employees whose experience is more than three years. Take data from the employee record table

```
CREATE PROCEDURE ThreeExpEmployees
AS
BEGIN
    SELECT *
    FROM employeeRecord
    WHERE EXP > 3
    ORDER BY EXP DESC;
END;
```

```
EXEC ThreeExpEmployees;
```

	EMP_ID	FIRST_NAME	LAST_NAME	GENDER	ROLE	DEPT	EXP	COUNTRY	CONTINENT	SALARY	EMP_RATING	MANAGER_ID	PROJ_ID
1	E001	Arthur	Black	M	PRESIDENT	ALL	20	USA	NORTH AMERICA	16500.00	5	NULL	NULL
2	E083	Patrick	Voltz	M	MANAGER	HEALTHCARE	15	USA	NORTH AMERICA	9500.00	5	E001	NULL
3	E103	Emily	Grove	F	MANAGER	FINANCE	14	CANADA	NORTH AMERICA	10500.00	4	E001	NULL
4	E428	Pete	Allen	M	MANAGER	AUTOMOTIVE	14	GERMANY	EUROPE	11000.00	4	E001	NULL



REQUEST 11

Write a query using stored functions in the project table to check whether the job profile assigned to each employee in the data science team matches the organization's set standard. :

- For an employee with experience less than or equal to 2 years assign 'JUNIOR DATA SCIENTIST'
- 2 to 5 years assign 'ASSOCIATE DATA SCIENTIST'
- 5 to 10 years assign 'SENIOR DATA SCIENTIST'
- 10 to 12 years assign 'LEAD DATA SCIENTIST'
- 12 to 16 years assign 'MANAGER'.

```
SELECT
    EMP_ID,
    FIRST_NAME,
    LAST_NAME,
    EXP,
    ROLE AS Current_Profile,
    dbo.fn_StandardJobProfile(EXP) AS Expected_Profile
FROM dataScienceTeam;
```

```
CREATE FUNCTION dbo.fn_StandardJobProfile (@exp INT)
RETURNS VARCHAR(50)
AS
BEGIN
    DECLARE @profile VARCHAR(50)

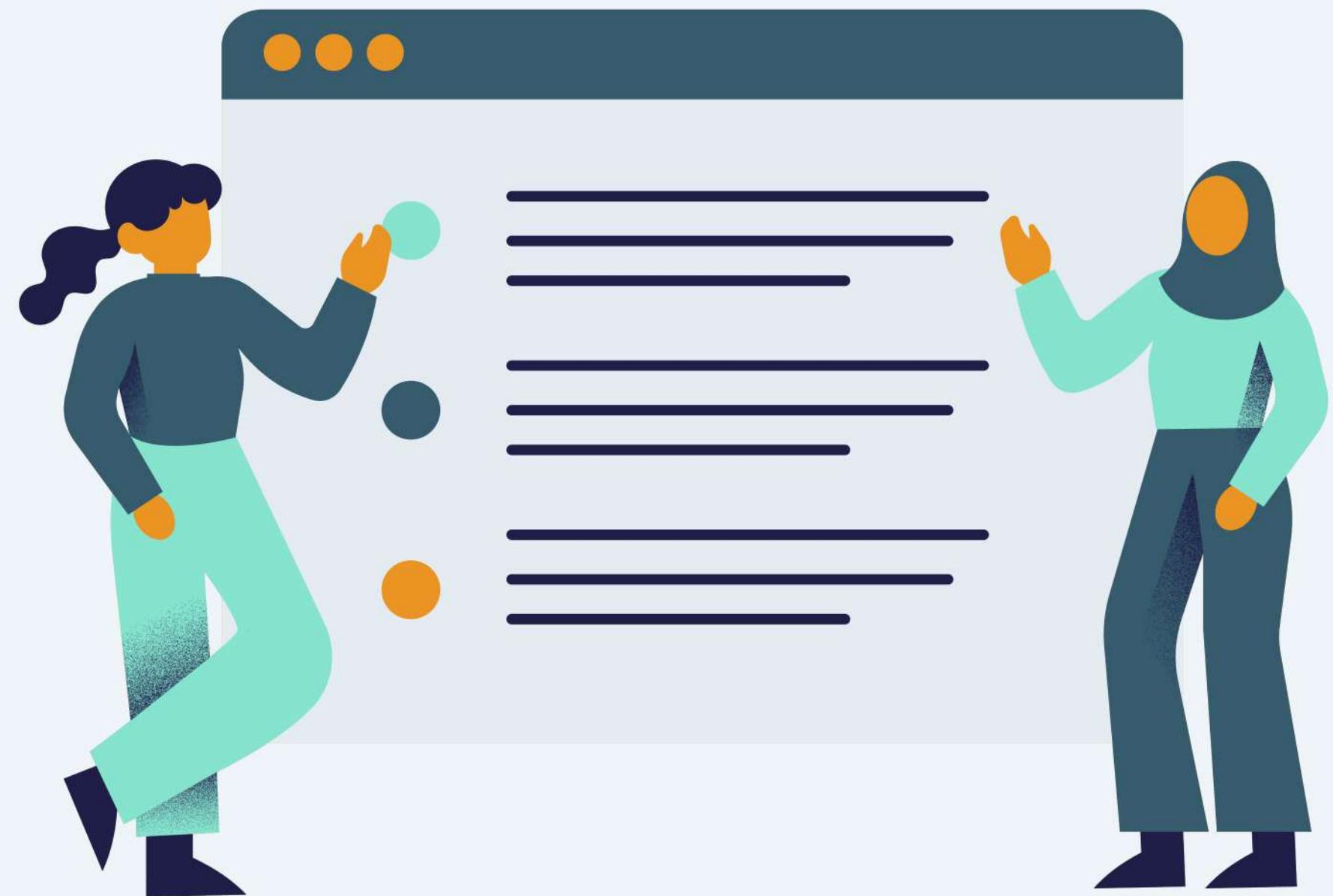
    IF @exp <= 2
        SET @profile = 'JUNIOR DATA SCIENTIST'
    ELSE IF @exp > 2 AND @exp <= 5
        SET @profile = 'ASSOCIATE DATA SCIENTIST'
    ELSE IF @exp > 5 AND @exp <= 10
        SET @profile = 'SENIOR DATA SCIENTIST'
    ELSE IF @exp > 10 AND @exp <= 12
        SET @profile = 'LEAD DATA SCIENTIST'
    ELSE IF @exp > 12 AND @exp <= 16
        SET @profile = 'MANAGER'
    ELSE
        SET @profile = 'N/A'

    RETURN @profile
END;
```



ADVANCED ANALYSIS

This section applies advanced techniques to solve complex analytical problems using structured blocks and row-level calculations. It includes generating temporary result sets for layered logic, assigning rankings within departments, calculating running totals, and comparing employee data within defined groups. These methods enable deep insights without relying on direct table relationships.



REQUEST 12

Write a query to list only those employees who have someone reporting to them. Also, show the number of reporters (including the President)

```
SELECT
    e.EMP_ID, e.FIRST_NAME, e.LAST_NAME, e.ROLE,
    COUNT(r.EMP_ID) AS Num_Reporters
    FROM employeeRecord e
    JOIN employeeRecord r
    ON e.EMP_ID = r.MANAGER_ID
    GROUP BY e.EMP_ID, e.FIRST_NAME, e.ROLE, e.LAST_NAME
    ORDER BY Num_Reporters DESC
```



	EMP_ID	FIRST_NAME	LAST_NAME	ROLE	Num_Reporters
1	E001	Arthur	Black	PRESIDENT	5
2	E083	Patrick	Voltz	MANAGER	3
3	E428	Pete	Allen	MANAGER	3
4	E583	Janet	Hale	MANAGER	3
5	E612	Tracy	Norris	MANAGER	2
6	E103	Emily	Grove	MANAGER	2



REQUEST 13

Write a query to assign ranks to each employee based on their experience. Take data from the employee record table.

```
SELECT EMP_ID, FIRST_NAME, LAST_NAME, ROLE, DEPT, EMP_RATING,  
EXP, RANK() OVER(ORDER BY EXP DESC) AS RANK_EXP  
FROM employeeRecord
```



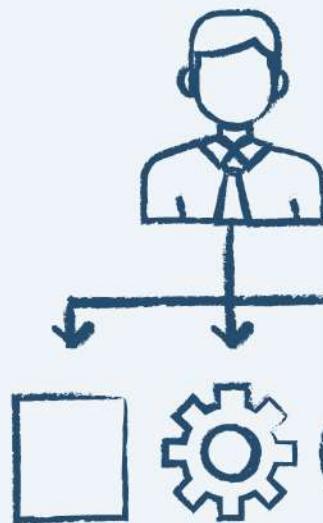
	EMP_ID	FIRST_NAME	LAST_NAME	ROLE	DEPT	EMP_RATING	EXP	RANK_EXP
	1	E001	Arthur	Black	PRESIDENT	ALL	5	20 1
	2	E083	Patrick	Voltz	MANAGER	HEALTHCARE	5	15 2
	3	E103	Emily	Grove	MANAGER	FINANCE	4	14 3
	4	E428	Pete	Allen	MANAGER	AUTOMOTIVE	4	14 3
	5	E583	Janet	Hale	MANAGER	RETAIL	2	14 3
	6	E612	Tracy	Nomis	MANAGER	RETAIL	4	13 6
	7	E010	William	Butler	LEAD DATA SCIENTIST	AUTOMOTIVE	2	12 7
	8	E005	Eric	Hoffman	LEAD DATA SCIENTIST	FINANCE	3	11 8



REQUEST 14

Write a query to display employee names along with the names of the projects they are assigned to.

```
SELECT  
e.EMP_ID,e.FIRST_NAME,e.LAST_NAME,p.PROJ_NAME  
FROM employeeRecord e  
INNER JOIN projectTable p  
ON e.PROJ_ID = p.PROJECT_ID;
```



	EMP_ID	FIRST_NAME	LAST_NAME	PROJ_NAME
1	E005	Eric	Hoffman	Fraud Detection
2	E010	William	Butler	Supply Chain Management
3	E052	Dianna	Wilson	Drug Discovery
4	E057	Dorothy	Wilson	Early Detection of Lung Cancer
5	E204	Karene	Nowak	Supply Chain Management
6	E245	Nian	Zhen	Market Basket Analysis
7	E403	Steve	Hoffman	Fraud Detection
8	E478	David	Smith	Market Basket Analysis
9	E505	Chad	Wilson	Drug Discovery
10	E532	Claire	Brennan	Supply Chain Management
11	E620	Katrina	Allen	Customer Sentiment Analysis



REQUEST 15

List employees, those who are not assigned to any project

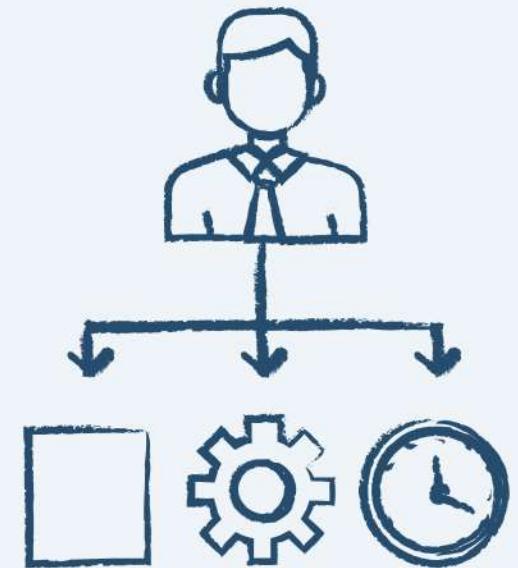
```
SELECT  
    e.EMP_ID, e.FIRST_NAME, e.LAST_NAME, p.PROJ_NAME  
FROM employeeRecord e  
LEFT JOIN projectTable p  
ON e.PROJ_ID = p.PROJECT_ID  
WHERE p.PROJ_NAME IS NULL
```

	EMP_ID	FIRST_NAME	LAST_NAME	PROJ_NAME
1	E001	Arthur	Black	NULL
2	E083	Patrick	Voltz	NULL
3	E103	Emily	Grove	NULL
4	E260	Roy	Collins	NULL
5	E428	Pete	Allen	NULL
6	E583	Janet	Hale	NULL
7	E612	Tracy	Nomis	NULL



REQUEST 16

List employee names, their project names, and their managers' names.



```
SELECT  
    e.FIRST_NAME AS Employee,  
    p.PROJ_NAME,  
    m.FIRST_NAME AS Manager  
FROM employeeRecord e  
JOIN projectTable p  
    ON e.PROJ_ID = p.PROJECT_ID  
LEFT JOIN employeeRecord m  
    ON e.MANAGER_ID = m.EMP_ID;
```

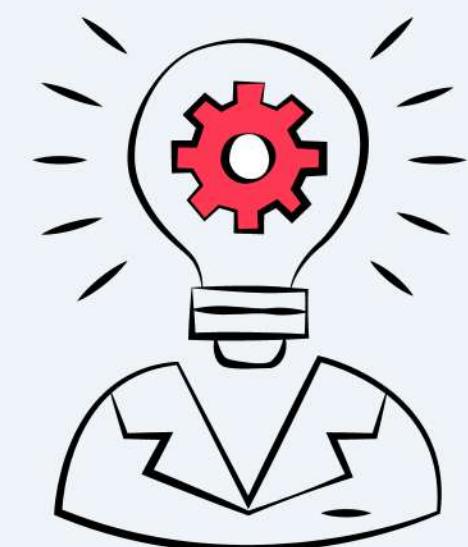
	Employee	PROJ_NAME	Manager
1	Eric	Fraud Detection	Emily
2	William	Supply Chain Management	Pete
3	Dianna	Drug Discovery	Patrick
4	Dorothy	Early Detection of Lung Cancer	Patrick
5	Karene	Supply Chain Management	Pete
6	Nian	Market Basket Analysis	Janet
7	Steve	Fraud Detection	Emily
8	David	Market Basket Analysis	Janet
9	Chad	Drug Discovery	Patrick
10	Claire	Supply Chain Management	Pete



REQUEST 17

Show employees who manage others but also report to another manager.

```
SELECT DISTINCT m.EMP_ID, m.FIRST_NAME, m.LAST_NAME  
FROM employeeRecord e  
JOIN employeeRecord m ON e.MANAGER_ID = m.EMP_ID  
WHERE m.MANAGER_ID IS NOT NULL;
```



	EMP_ID	FIRST_NAME	LAST_NAME
1	E083	Patrick	Voltz
2	E103	Emily	Grove
3	E428	Pete	Allen
4	E583	Janet	Hale
5	E612	Tracy	Norris



REQUEST 18

List projects that have only one employee assigned.

```
SELECT p.PROJ_NAME, COUNT(e.EMP_ID) AS Assigned_Employees  
FROM projectTable p  
JOIN employeeRecord e ON p.PROJECT_ID = e.PROJ_ID  
GROUP BY p.PROJ_NAME  
HAVING COUNT(e.EMP_ID) = 1;
```

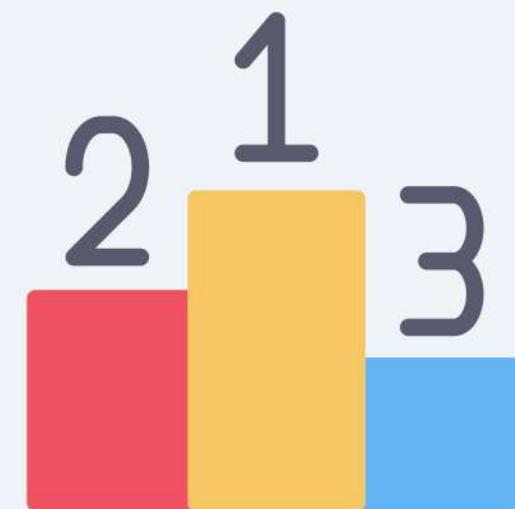


	PROJ_NAME	Assigned_Employees
1	Early Detection of Lung Cancer	1



REQUEST 19

Find top 3 highest paid employees per department



```
WITH RankedSalaries AS (
    SELECT *,
        RANK() OVER (PARTITION BY DEPT ORDER BY SALARY DESC) AS rk
    FROM employeeRecord
)
SELECT EMP_ID, FIRST_NAME, DEPT, SALARY
FROM RankedSalaries
WHERE rk <= 3
ORDER BY DEPT ASC
```

	EMP_ID	FIRST_NAME	DEPT	SALARY
1	E001	Arthur	ALL	16500.00
2	E428	Pete	AUTOMOTIVE	11000.00
3	E010	William	AUTOMOTIVE	9000.00
4	E204	Karene	AUTOMOTIVE	7500.00
5	E103	Emily	FINANCE	10500.00
6	E005	Eric	FINANCE	8500.00
7	E403	Steve	FINANCE	5000.00
8	E083	Patrick	HEALTHCARE	9500.00
9	E057	Dorothy	HEALTHCARE	7700.00
10	E052	Dianna	HEALTHCARE	5500.00



CONCLUSION!

This project demonstrates how structured SQL techniques can transform raw employee and project data into meaningful insights. By applying foundational querying, schema-level optimizations, and advanced analytical logic, the solution enhances data accessibility, accuracy, and performance. It highlights the role of SQL not just as a data extraction tool, but as a strategic asset for decision-making and organizational efficiency.





SCIENCEQTECH

THANK YOU!
WE ARE GLAD
YOU ARE HERE.

SQL PROJECT: END-TO-END
DATA EXPLORATION AND
OPTIMIZATION

