



Task 1:

HR Analysis

SQL Internship at Psyliq

By [Utsav Raj](#)



1. Retrieve the total number of employees in the dataset.

➡ SELECT count(EmployeeID) FROM general_data;

The screenshot shows a SQL query editor interface. At the top, there is a toolbar with various icons for file operations, execution, and viewing. Below the toolbar, the query text is displayed in a monospace font. The first line is a comment, and the second line is the SQL query. Below the query, there is a 'Result Grid' section. The grid has two columns: the first column contains the SQL expression 'count(EmployeeID)' and the second column contains the result value '4382'.

```
1  #Retrieve the total number of employees in the dataset.  
2 • SELECT count(EmployeeID) FROM general_data;
```

	count(EmployeeID)
▶	4382



2. List all unique job roles in the dataset.

→ `select distinct(JobRole) FROM general_data;`

The screenshot shows a data analysis tool interface. At the top, there is a toolbar with various icons and a dropdown menu set to "Limit to 50000 rows". Below the toolbar, a SQL query is entered in a text area:

```
4 #List all unique job roles in the dataset.  
5 • select distinct(JobRole) FROM general_data;  
6  
7  
8
```

Below the query editor, there is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The results are displayed in a table with one column, "JobRole". The table contains the following unique job roles:

JobRole
Healthcare Representative
Research Scientist
Sales Executive
Human Resources
Research Director
Laboratory Technician
Manufacturing Director
Sales Representative
Manager



3. Find the average age of employees.

➡ select avg(age) as average_age FROM general_data;

The screenshot shows a SQL query editor window. The query is: `select avg(age) as average_age FROM general_data;`. Below the query, the results are displayed in a table with one column, `average_age`, and one row containing the value `36.9334`.

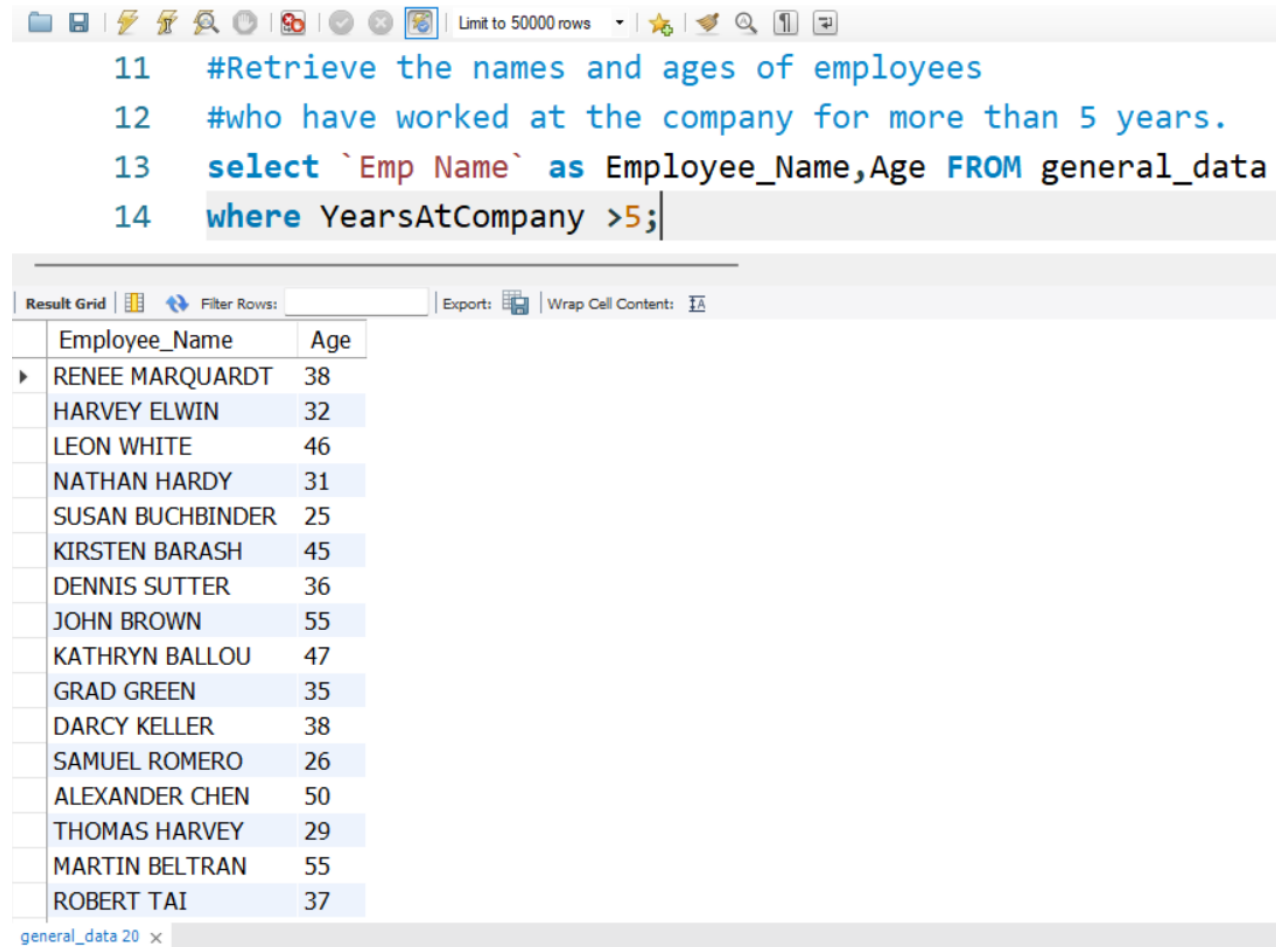
```
7
8  #Find the average age of employees.
9 • select avg(age) as average_age FROM general_data;
10
```

average_age
36.9334



4.Retrieve the names and ages of employees who have worked at the company for more than 5 years.

➡ select `Emp Name` as Employee_Name,Age FROM general_data where YearsAtCompany >5;



```
11 #Retrieve the names and ages of employees
12 #who have worked at the company for more than 5 years.
13 select `Emp Name` as Employee_Name,Age FROM general_data
14 where YearsAtCompany >5;
```

Employee_Name	Age
RENEE MARQUARDT	38
HARVEY ELWIN	32
LEON WHITE	46
NATHAN HARDY	31
SUSAN BUCHBINDER	25
KIRSTEN BARASH	45
DENNIS SUTTER	36
JOHN BROWN	55
KATHRYN BALLOU	47
GRAD GREEN	35
DARCY KELLER	38
SAMUEL ROMERO	26
ALEXANDER CHEN	50
THOMAS HARVEY	29
MARTIN BELTRAN	55
ROBERT TAI	37



5. Get a count of employees grouped by their department.

➡ select Department, count(*) as Employee_count FROM
general_data group by Department;

The screenshot shows a SQL IDE interface. The top toolbar includes icons for file operations, execution, and a 'Limit to 50000 rows' dropdown. The query editor contains the following SQL code:

```
17  
18 #Get a count of employees grouped by their department.  
19 • select Department, count(*) as Employee_count FROM general_data  
20 group by Department;  
21
```

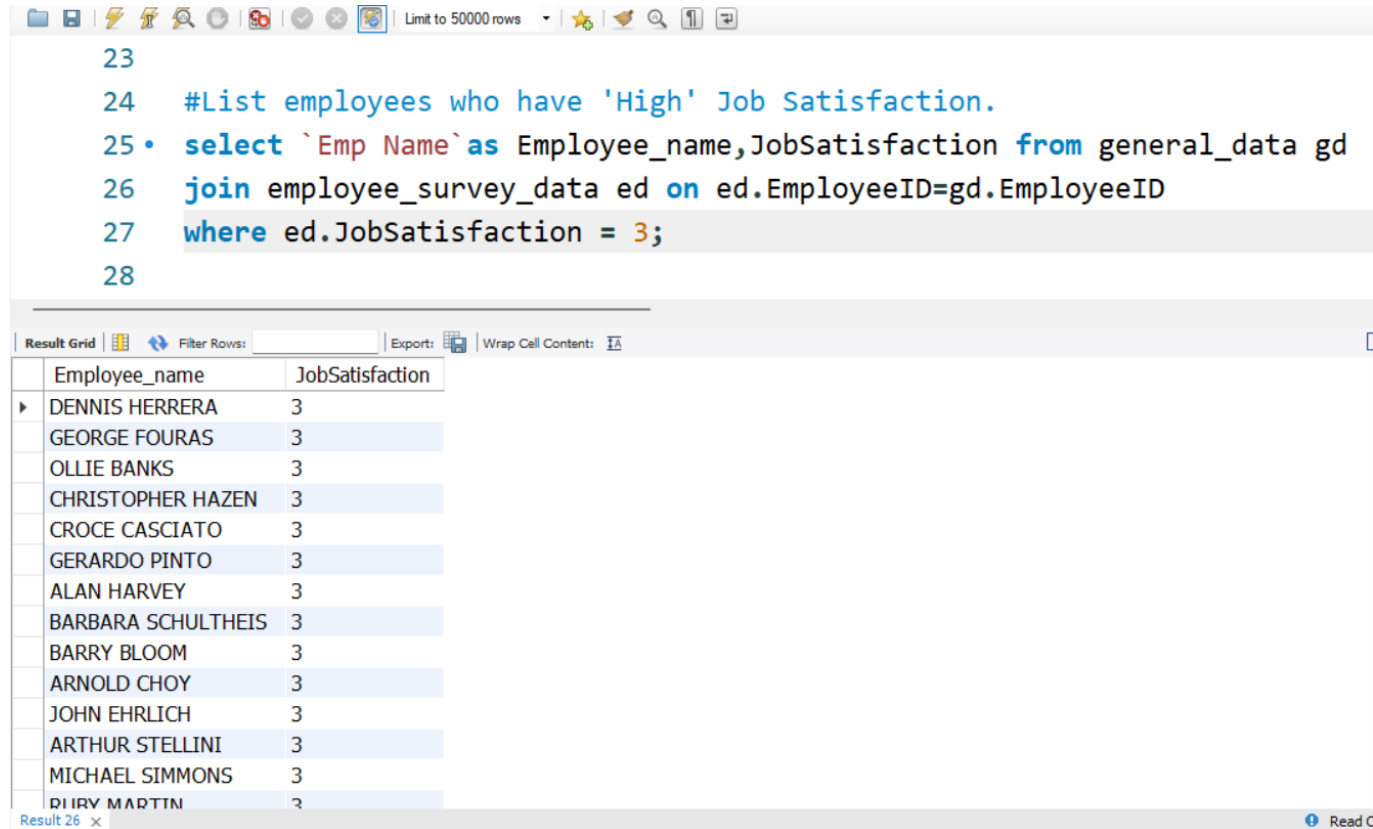
Below the query editor is the 'Result Grid' section. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The results are displayed in a table with two columns: 'Department' and 'Employee_count'.

Department	Employee_count
Sales	1330
Research & Developm...	2865
Human Resources	187



6. List employees who have 'High' Job Satisfaction.

→ select `Emp Name` as Employee_name, JobSatisfaction from
general_data gd join employee_survey_data ed on
ed.EmployeeID=gd.EmployeeID where ed.JobSatisfaction = 3;



```
23
24 #List employees who have 'High' Job Satisfaction.
25 • select `Emp Name` as Employee_name, JobSatisfaction from general_data gd
26 join employee_survey_data ed on ed.EmployeeID=gd.EmployeeID
27 where ed.JobSatisfaction = 3;
28
```

Employee_name	JobSatisfaction
DENNIS HERRERA	3
GEORGE FOURAS	3
OLLIE BANKS	3
CHRISTOPHER HAZEN	3
CROCE CASCIATO	3
GERARDO PINTO	3
ALAN HARVEY	3
BARBARA SCHULTHEIS	3
BARRY BLOOM	3
ARNOLD CHOY	3
JOHN EHRLICH	3
ARTHUR STELLINI	3
MICHAEL SIMMONS	3
DAVID MARTIN	3



7. Find the highest Monthly Income in the dataset.

→ select max(MonthlyIncome) as Highest_monthly_income from general_data;

The screenshot shows a SQL query editor window with a toolbar at the top. The toolbar includes icons for file operations, execution, and a dropdown menu set to 'Limit to 50000 rows'. The query text is as follows:

```
28  
29 #Find the highest Monthly Income in the dataset.  
30 • select max(MonthlyIncome) as Highest_monthly_income from general_data;  
31
```

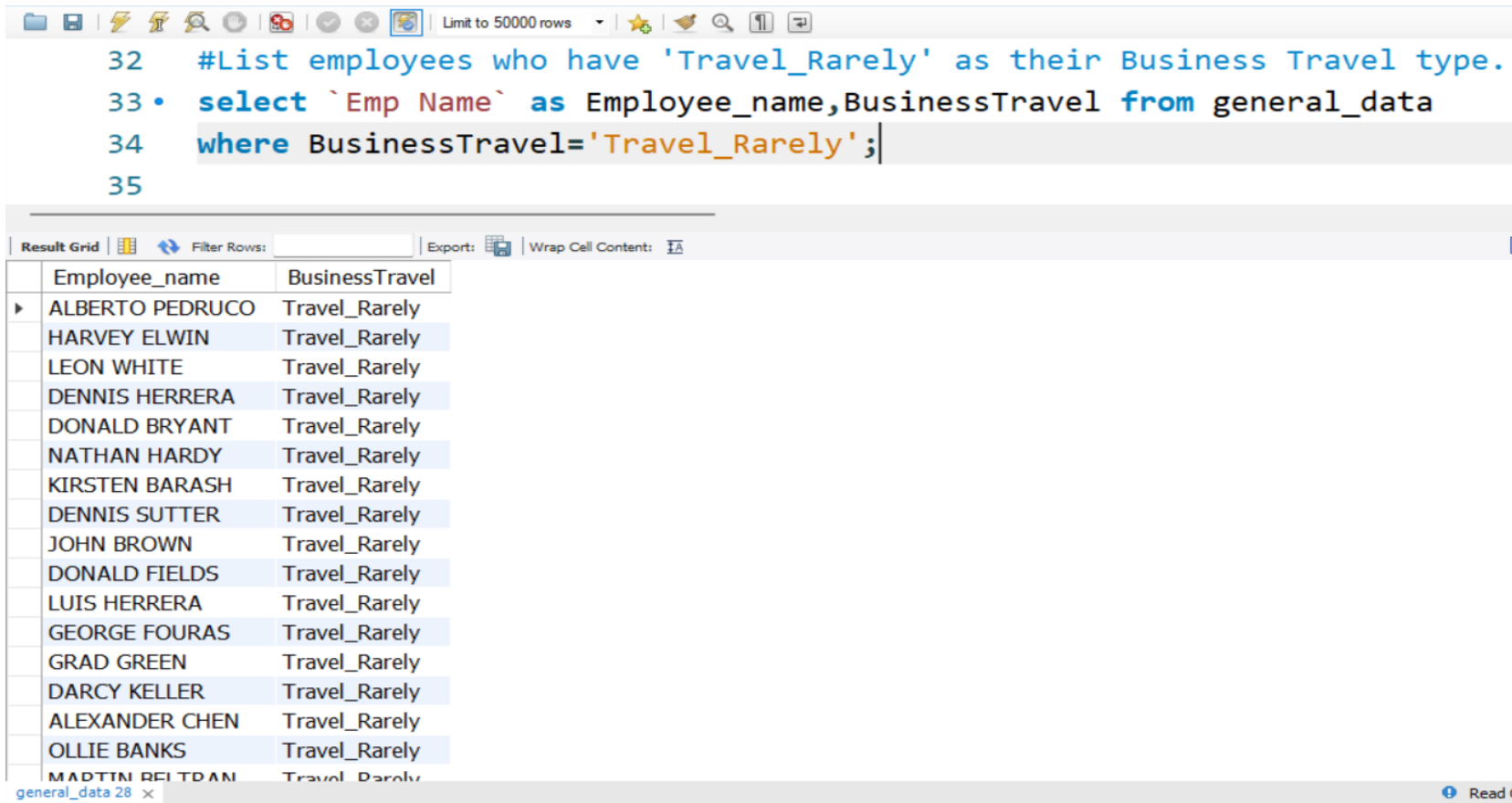
Below the query editor is a 'Result Grid' section. It has a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The grid displays the following data:

Highest_monthly_income
199990



8. List employees who have 'Travel_Rarely' as their Business Travel type.

→ select `Emp Name` as Employee_name,BusinessTravel from
general_data where BusinessTravel='Travel_Rarely';



```
32 #List employees who have 'Travel_Rarely' as their Business Travel type.
33 • select `Emp Name` as Employee_name,BusinessTravel from general_data
34 where BusinessTravel='Travel_Rarely';
35
```

Employee_name	BusinessTravel
ALBERTO PEDRUCO	Travel_Rarely
HARVEY ELWIN	Travel_Rarely
LEON WHITE	Travel_Rarely
DENNIS HERRERA	Travel_Rarely
DONALD BRYANT	Travel_Rarely
NATHAN HARDY	Travel_Rarely
KIRSTEN BARASH	Travel_Rarely
DENNIS SUTTER	Travel_Rarely
JOHN BROWN	Travel_Rarely
DONALD FIELDS	Travel_Rarely
LUIS HERRERA	Travel_Rarely
GEORGE FOURAS	Travel_Rarely
GRAD GREEN	Travel_Rarely
DARCY KELLER	Travel_Rarely
ALEXANDER CHEN	Travel_Rarely
OLLIE BANKS	Travel_Rarely
MARTIN REITMAN	Travel_Rarely



9. Retrieve the distinct Marital Status categories in the dataset.

➡ select distinct(MaritalStatus) from general_data;

```
36
37 #Retrieve the distinct MaritalStatus categories in the dataset.
38 • select distinct(MaritalStatus) from general_data;
```

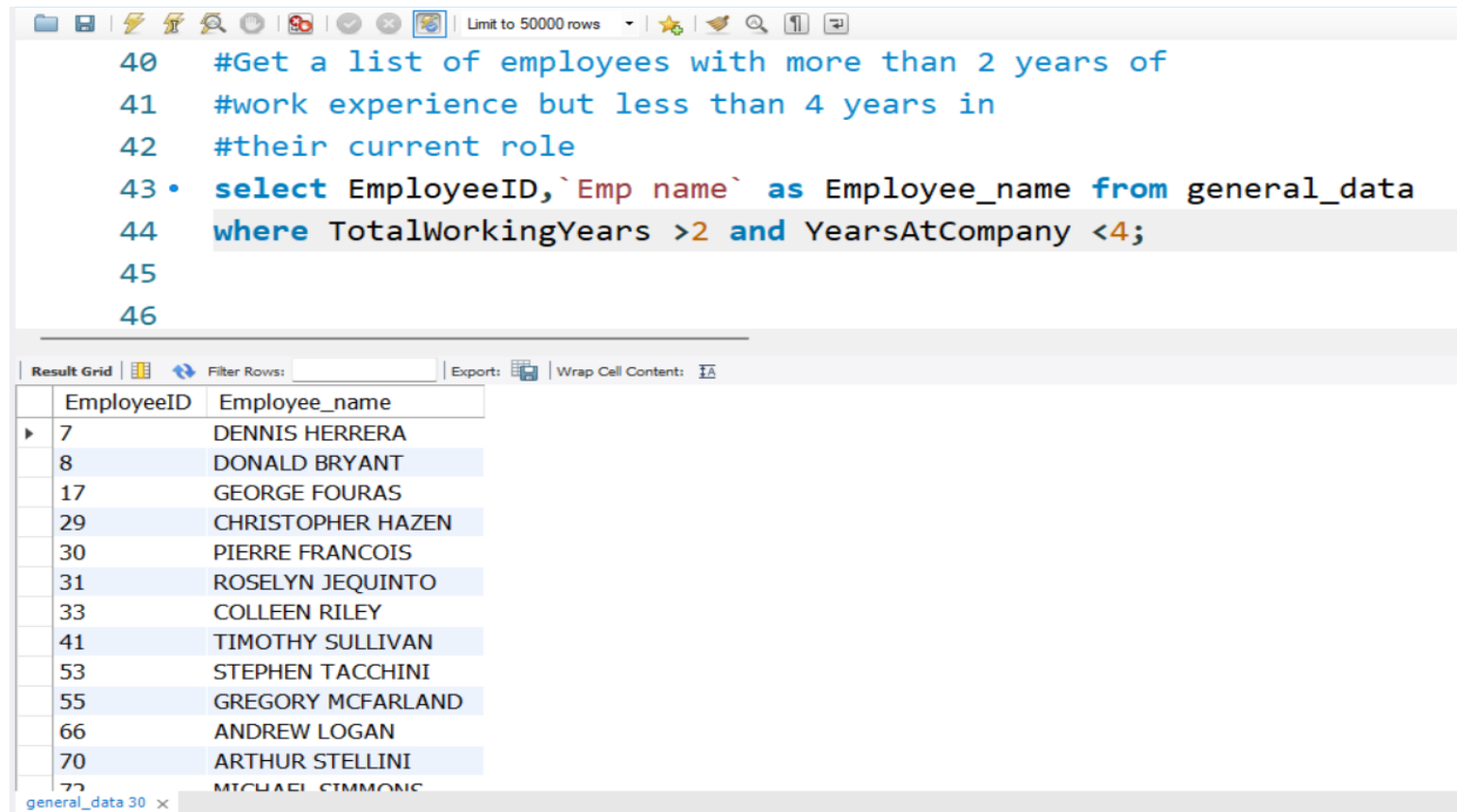
Result Grid		Filter Rows:	Export:	Wrap Cell Content:
MaritalStatus				
Married				
Single				
Divorced				

Result 29 x



10. Get a list of employees with more than 2 years of work experience but less than 4 years in their current role.

➡ select EmployeeID, `Emp name` as Employee_name from general_data where TotalWorkingYears >2 and YearsAtCompany <4;



```
40 #Get a list of employees with more than 2 years of
41 #work experience but less than 4 years in
42 #their current role
43 • select EmployeeID, `Emp name` as Employee_name from general_data
44 where TotalWorkingYears >2 and YearsAtCompany <4;
45
46
```

EmployeeID	Employee_name
7	DENNIS HERRERA
8	DONALD BRYANT
17	GEORGE FOURAS
29	CHRISTOPHER HAZEN
30	PIERRE FRANCOIS
31	ROSELYN JEQUINTO
33	COLLEEN RILEY
41	TIMOTHY SULLIVAN
53	STEPHEN TACCHINI
55	GREGORY MCFARLAND
66	ANDREW LOGAN
70	ARTHUR STELLINI
77	MICHAEL STIMMONS

general_data 30 x



11. List employees who have changed their job roles within the company (Job Level and Job Role differ from their previous job).

➡ select * from general_data as current join general_data as previous on current.EmployeeID=previous.EmployeeID where (current.JobRole<>previous.JobRole or previous.EmployeeID is null) or (Current.JobLevel<>previous.JobLevel or previous.EmployeeID is null);

The screenshot shows a SQL IDE window with a query editor and a result grid. The query editor contains the following SQL code:

```
47 #List employees who have changed their job roles within
48 #the company (JobLevel and JobRole differ from their previous job).
49 • select * from general_data as current
50 join general_data as previous on current.EmployeeID=previous.EmployeeID
51 where (current.JobRole<>previous.JobRole or previous.EmployeeID is null)
52 or (Current.JobLevel<>previous.JobLevel or previous.EmployeeID is null);
53
```

The result grid below the query editor shows the following columns:

Emp Name	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeID
----------	-----	-----------	----------------	------------	------------------	-----------	----------------	---------------	------------



12. Find the average distance from home for employees in each department.

→ select Department, round(avg(DistanceFromHome),2) as Avg_distance from general_data group by Department;

The screenshot shows a SQL query editor window with a toolbar at the top. The query is as follows:

```
52  
53 #Find the average distance from home for employees in each department.  
54 • select Department, round(avg(DistanceFromHome),2) as Avg_distance  
55 from general_data group by Department;  
56  
57
```

Below the query editor is the 'Result Grid' section, which displays the results of the query. It includes a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' checkbox. The results are shown in a table with two columns: 'Department' and 'Avg_distance'.

Department	Avg_distance
Sales	9.24
Research & Developm...	9.24
Human Resources	8.29

At the bottom of the window, there is a tab labeled 'Result 33' and a 'Rea' button.



13. Retrieve the top 5 employees with the highest Monthly Income.

➡ select EmployeeID, `Emp Name` as 'Top 5 Highest Paid Employee', MonthlyIncome from general_data order by MonthlyIncome desc limit 5;

```
57
58 #Retrieve the top 5 employees with the highest MonthlyIncome.
59 • select EmployeeID, `Emp Name` as 'Top 5 Highest Paid Employee',
60   MonthlyIncome from general_data
61   order by MonthlyIncome desc limit 5;
62
63
```

Result Grid			
Filter Rows: <input type="text"/> Export: Wrap Cell Content: Fetch rows:			
	EmployeeID	Top 5 Highest Paid Employee	MonthlyIncome
▶	386	KEVIN LABANOWSKI	199990
	1856	DAVID KUCIA	199990
	3326	LAWRENCE LAU	199990
	942	KEVIN MCNAUGHTON	199730
	3882	SHANNON STABILE	199730



14. Calculate the percentage of employees who have had a promotion in the last year.

→ select count(*) as Total_employees, count(case when YearsSinceLastPromotion<=1 then 1 end) as Count_promotion_last_year, (count(case when YearsSinceLastPromotion<=1 then 1 end)*100/count(*)) as Percentage from general_data;

```
63
64 #Calculate the percentage of employees who have had a promotion
65 #in the last year.
66 • select count(*) as Total_employees,
67    count(case when YearsSinceLastPromotion<=1 then 1 end) as
68    Count_promotion_last_year,
69    (count(case when YearsSinceLastPromotion<=1 then 1 end)*100/count(*))
70    as Percentage from general_data;
71
72
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	Total_employees	Count_promotion_last_year	Percentage
▶	4382	2792	63.7152



15. List the employees with the highest and lowest Environment Satisfaction.

→ `select gd.EmployeeID, `Emp Name` as Employee_name,
EnvironmentSatisfaction from general_data gd join employee_survey_data as ed on
ed.EmployeeID=gd.EmployeeID where ed.EnvironmentSatisfaction in(select
max(EnvironmentSatisfaction) from employee_survey_data union select
min(EnvironmentSatisfaction) from employee_survey_data);`

The screenshot shows a SQL query editor window with a toolbar at the top. The query is as follows:

```
74 #List the employees with the highest and lowest EnvironmentSatisfaction.  
75 • select gd.EmployeeID, `Emp Name` as Employee_name, EnvironmentSatisfaction  
76 from general_data gd  
77 join employee_survey_data as ed on ed.EmployeeID=gd.EmployeeID  
78 where ed.EnvironmentSatisfaction in  
79 (select max(EnvironmentSatisfaction) from employee_survey_data union  
80 select min(EnvironmentSatisfaction) from employee_survey_data);
```

Below the query editor is a 'Result Grid' window showing the results of the query. The grid has three columns: EmployeeID, Employee_name, and EnvironmentSatisfaction. The results are as follows:

EmployeeID	Employee_name	EnvironmentSatisfaction
4	RENEE MARQUARDT	4
5	HARVEY ELWIN	4
7	DENNIS HERRERA	1
8	DONALD BRYANT	1
13	JOHN BROWN	4
14	KATHRYN BALLOU	1
15	DONALD FIELDS	4
17	GEORGE FOURAS	4
18	MARTIN LALOR JR	1
20	DARCY KELLER	1
22	ALEXANDER CHEN	1
27	LORI BORCHI	1

The bottom of the window shows 'Result 41' and a 'Read On' button.



16. Find the employees who have the same Job Role and Marital Status.

select EmployeeId, `Emp Name` as Employee_Name, JobRole, MaritalStatus from general_data as a where exists (select 1 from general_data as b where a.EmployeeId<>b.EmployeeId and a.JobRole like b.JobRole and a.MaritalStatus like b.MaritalStatus) order by JobRole,MaritalStatus,EmployeeID;

The screenshot shows a SQL IDE window with a query editor and a results grid. The query is as follows:

```
83 #Find the employees who have the same JobRole and MaritalStatus.
84 • select EmployeeId, `Emp Name` as Employee_Name, JobRole, MaritalStatus
85 from general_data as a where exists (select 1 from general_data as b
86 where a.EmployeeId<>b.EmployeeId and a.JobRole like b.JobRole and
87 a.MaritalStatus like b.MaritalStatus) order by JobRole,MaritalStatus,
88 EmployeeID;
89
```

The results grid displays the following data:

EmployeeId	Employee_Name	JobRole	MaritalStatus
64	REGINA GOMEZ	Healthcare Representative	Divorced
83	CRISTINA REYES	Healthcare Representative	Divorced
147	ANN MANNIX	Healthcare Representative	Divorced
152	RYAN CREAM	Healthcare Representative	Divorced
161	LINDA ROSS	Healthcare Representative	Divorced
203	DANIEL ARMENTA	Healthcare Representative	Divorced
234	MICHAEL DALY	Healthcare Representative	Divorced
262	FUAD SWEISS	Healthcare Representative	Divorced
300	MANUEL PEGUEROS	Healthcare Representative	Divorced
368	ANDRE ANDREWS SR	Healthcare Representative	Divorced
372	EDWARD CHEUNG	Healthcare Representative	Divorced
400	FRANK KOSTA	Healthcare Representative	Divorced

The results grid also includes a 'Filter Rows' section and an 'Export' button. The status bar at the bottom indicates 'general_data 43 x' and 'Read On'.



17. List the employees with the highest Total Working Years who also have a Performance Rating of 4.

```
select gd.EmployeeID, gd.`Emp name` as Employee_name from general_data gd
join manager_survey_data md on gd.EmployeeID = md.EmployeeID where
md.PerformanceRating=4 and gd.TotalWorkingYears= (select
max(TotalWorkingYears) from general_data where EmployeeId in (select
EmployeeId from manager_survey_data where PerformanceRating =4))
```

```
91 #List the employees with the highest TotalWorkingYears
92 #who also have a PerformanceRating of 4.
93 • select gd.EmployeeID, gd.`Emp name` as Employee_name
94 from general_data gd join manager_survey_data md on
95 gd.EmployeeID = md.EmployeeID where md.PerformanceRating=4 and
96 gd.TotalWorkingYears= (select max(TotalWorkingYears) from general_data
97 where EmployeeId in (select EmployeeId from manager_survey_data where
98 PerformanceRating =4))
99
```

EmployeeID	Employee_name
786	STEVEN SETO
2256	SHARON LEGENZA
3726	EMILY MURASE



18. Calculate the average Age and Job Satisfaction for each Business Travel type.

➡ select BusinessTravel, Avg(age) as Average_age, avg(ed.JobSatisfaction) as Average_Job_satisfaction from general_data gd join employee_survey_data ed on gd.EmployeeId=ed.EmployeeId group by BusinessTravel;

The screenshot shows a SQL query editor window with a toolbar at the top. The query is as follows:

```
101
102  #Calculate the average Age and JobSatisfaction
103  #for each BusinessTravel type.
104  select BusinessTravel, Avg(age) as Average_age, avg(ed.JobSatisfaction)
105  as Average_Job_satisfaction from general_data gd
106  join employee_survey_data ed on gd.EmployeeId=ed.EmployeeId
107  group by BusinessTravel;
```

Below the query editor is the 'Result Grid' showing the output of the query. It has columns for BusinessTravel, Average_age, and Average_Job_satisfaction. The results are as follows:

BusinessTravel	Average_age	Average_Job_satisfaction
Travel_Rarely	37.1036	2.7001
Travel_Frequently	36.4722	2.7886
Non-Travel	36.5386	2.7795



19. Retrieve the most common Education Field among employees.

➡ select EducationField,count(EducationField) as most_common_field from general_data group by EducationField order by most_common_field desc limit 1;

The screenshot shows a SQL query editor with a toolbar at the top containing icons for file operations, search, and execution. The query is as follows:

```
110
111 #Retrieve the most common EducationField among employees.
112 • select EducationField,count(EducationField) as most_common_field
113 from general_data group by EducationField
114 order by most_common_field desc limit 1;
115
```

Below the query editor is a results pane with a toolbar for filtering, exporting, and wrapping content. It displays a single row of results in a table format:

EducationField	most_common_field
Life Sciences	1806



20. List the employees who have worked for the company the longest but haven't had a promotion.



select EmployeeId, `Emp Name` as Employee_name, YearsAtCompany, YearsSinceLastPromotion from general_data where YearsAtCompany in (select max(YearsAtCompany) from general_data where YearsSinceLastPromotion=0) and YearsSinceLastPromotion=0;

```
117  #List the employees who have worked for the company
118  #the longest but haven't had a promotion.
119 • select EmployeeId, `Emp Name` as Employee_name, YearsAtCompany,
120     YearsSinceLastPromotion from general_data
121  where YearsAtCompany in (select max(YearsAtCompany) from general_data
122     where YearsSinceLastPromotion=0) and YearsSinceLastPromotion=0;
123
```

Result Grid Filter Rows: Export: Wrap Cell Content:				
	EmployeeId	Employee_name	YearsAtCompany	YearsSinceLastPromotion
▶	705	DENISE FLAHERTY	33	0
	2175	HERMAN WOO	33	0
	3645	YU-RUO WANG	33	0



Thank You

SQL Internship at Psyliq

By [Utsav Raj](#)