

HR DATA ASSESSMENT

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

INPUT: `SELECT count(EmployeeID) FROM employee_survey_data;`

OUTPUT: 4410

SCREENSHOT:

The screenshot shows a SQL query editor with the following query:

```
1 • SELECT count(EmployeeID) FROM employee_survey_data;
```

Below the query editor, the results are displayed in a table:

count(EmployeeID)
4410

The second screenshot shows the 'Output' pane with the following table:

#	Time	Action	Message
✓ 10	19:37:24	SELECT * FROM internship.employee_survey_data LIMIT 0, 1000	1000 row(s) returned
✓ 11	19:45:23	SELECT count(EmployeeID) FROM employee_survey_data LIMIT 0, 1000	1 row(s) returned

2. List all unique job roles in the dataset.

INPUT: SELECT distinct JobRole FROM general_data;

OUTPUT:

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



SCREENSHOT:


2 •

`SELECT distinct JobRole FROM general_data;`

<

Result Grid

  Filter Rows:


Export: 

Wrap Cell Content: [IA](#)

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

general_data 4 x

Output

 Action Output

#	Time	Action
✓ 1	19:51:43	SELECT distinct JobRole FROM general_data LIMIT 0, 1000

3. Find the average age of employees.

INPUT: SELECT avg(AGE) FROM general_data;

OUTPUT: 36.9334

SCREENSHOT:

The screenshot shows a SQL query editor with the query `SELECT avg(AGE) FROM general_data;` entered. Below the editor, the results are displayed in a table with one column, `avg(AGE)`, and one row containing the value `36.9334`. The interface includes a toolbar with options like 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. At the bottom, there is a section for 'Output' and 'Action Output', showing a log of the executed query with a green checkmark indicating success.

```
3 • SELECT avg(AGE) FROM general_data;
```

	avg(AGE)
▶	36.9334

Result 6 x

Output

Action Output

#	Time	Action
1	19:56:35	SELECT avg(AGE) FROM general_data LIMIT 0, 1000

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

INPUT: SELECT AGE FROM general_data
WHERE TotalWorkingYears >5;

OUTPUT:

31
38
32
46
29
31
25
45
36
55

SCREENSHOT:

4 •
5

SELECT AGE FROM general_data

WHERE TotalWorkingYears >5;

<

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Fetch rows:

	AGE
▶	31
	38
	32
	46
	29
	31
	25
	45
	36

general_data 7

×

Output

Action Output

	#	Time	Action
✓	1	20:04:45	SELECT AGE FROM general_data WHERE TotalWorkingYears >5 LIMIT 0, 1000

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

INPUT: SELECT count(EmployeeID) FROM general_data
GROUP BY Department;

OUTPUT:

1330
2865
187

SCREENSHOT:

The screenshot shows a SQL query editor with the following code:

```
5 • SELECT count(EmployeeID) FROM general_data
6 GROUP BY Department;
```

Below the editor is a toolbar with options: Result Grid, Filter Rows, Export, and Wrap Cell Content. The Result Grid shows the following data:

count(EmployeeID)
1330
2865
187

At the bottom, there is a section for "Result 8" and "Output". The "Output" section shows the execution of the query:

#	Time	Action
1	20:21:33	SELECT count(EmployeeID) FROM general_data GROUP BY Department LIMIT 0, 1000

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

INPUT: SELECT * FROM employee_survey_data
WHERE JobSatisfaction=3;

OUTPUT:

7	1	3	1
17	4	3	4
23	3	3	2
24	2	3	3
29	4	3	1
39	3	3	3
44	4	3	3
49	4	3	2
51	4	3	3
59	3	3	3

SCREENSHOT:

The screenshot displays a database query interface. At the top, the SQL query is entered in a text area:

```
6 • SELECT * FROM employee_survey_data
7 WHERE JobSatisfaction=3;
8
9
```

Below the query editor, there is a toolbar with options: "Result Grid", "Filter Rows:", "Export:", "Wrap Cell Content:", and "Fetch rows:". The "Result Grid" is active, showing a table with the following data:

	EmployeeID	EnvironmentSatisfaction	JobSatisfaction	WorkLifeBalance
▶	7	1	3	1
	17	4	3	4
	23	3	3	2
	24	2	3	3
	29	4	3	1
	39	3	3	3
	44	4	3	3
	49	4	3	2
	51	4	3	3
	59	3	3	3

Below the table, there is a tab labeled "employee_survey_data 14". At the bottom, there is an "Output" section with a table showing the execution details:

#	Time	Action	Message
✓ 1	21:13:57	SELECT * FROM employee_survey_data WHERE JobSatisfaction=3 LIMIT 0, 1000	1000 row(s) returned

7. Find the highest Monthly Income in the dataset.

INPUT: SELECT max(MonthlyIncome) FROM general_data;

OUTPUT: 199990

SCREENSHOT:

The screenshot displays a database query interface. At the top, a query editor shows the SQL statement: `SELECT max(MonthlyIncome) FROM general_data;`. Below the editor, a toolbar includes a 'Result Grid' icon, a 'Filter Rows' input field, an 'Export' button, and a 'Wrap Cell Content' button. The 'Result Grid' is active, showing a single column with the header `max(MonthlyIncome)` and one row containing the value `199990`. Below the result grid, a tab labeled 'Result 9' is visible. Underneath, an 'Output' section shows a table with the following data:

#	Time	Action	Message
1	21:03:19	SELECT max(MonthlyIncome) FROM general_data LIMIT 0, 1000	1 row(s) returned

8. List employees who have 'Travel_Rarely' as their BusinessTravel type.

INPUT: SELECT * FROM general_data
WHERE BusinessTravel='Travel_Rarely';

OUTPUT:

51	No	Travel_Rarely	Sales	6	2	Life Sciences
32	No	Travel_Rarely	Research & Development	10	1	Medical
46	No	Travel_Rarely	Research & Development	8	3	Life Sciences
28	Yes	Travel_Rarely	Research & Development	11	2	Medical
29	No	Travel_Rarely	Research & Development	18	3	Life Sciences
31	No	Travel_Rarely	Research & Development	1	3	Life Sciences
45	No	Travel_Rarely	Research & Development	17	2	Medical
36	No	Travel_Rarely	Research & Development	28	1	Life Sciences
55	No	Travel_Rarely	Research & Development	14	4	Life Sciences
28	No	Travel_Rarely	Research & Development	1	3	Life Sciences

SCREENSHOT:

```
8 • SELECT * FROM general_data
9 WHERE BusinessTravel='Travel_Rarely';
```

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

	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeID	Gender	Job
▶	51	No	Travel_Rarely	Sales	6	2	Life Sciences	1	1	Female	1
	32	No	Travel_Rarely	Research & Development	10	1	Medical	1	5	Male	1
	46	No	Travel_Rarely	Research & Development	8	3	Life Sciences	1	6	Female	4
	28	Yes	Travel_Rarely	Research & Development	11	2	Medical	1	7	Male	2
	29	No	Travel_Rarely	Research & Development	18	3	Life Sciences	1	8	Male	2
	31	No	Travel_Rarely	Research & Development	1	3	Life Sciences	1	9	Male	3
	45	No	Travel_Rarely	Research & Development	17	2	Medical	1	11	Male	2
	36	No	Travel_Rarely	Research & Development	28	1	Life Sciences	1	12	Male	1

general_data 11 x

Output

Action Output

#	Time	Action	Message
✓ 1	21:08:03	SELECT * FROM general_data WHERE BusinessTravel='Travel_Rarely' LIMIT 0, 1000	1000 row(s) returned

9. Retrieve the distinct MaritalStatus categories in the dataset.

INPUT: SELECT distinct MaritalStatus FROM general_data;

OUTPUT:

Married
Single
Divorced

SCREENSHOT:

The screenshot shows a SQL query execution interface. At the top, the query `SELECT distinct MaritalStatus FROM general_data;` is entered in a text box. Below the query, there is a toolbar with options like "Result Grid", "Filter Rows", "Export", and "Wrap Cell Content". The "Result Grid" is selected, and it displays the output of the query in a table with three rows: "Married", "Single", and "Divorced". Below the result grid, there is a section labeled "Output" which shows the execution details. It includes a table with columns "#", "Time", "Action", and "Message". The table contains one row indicating that the query was executed successfully at 21:19:42, returning 3 rows.

#	Time	Action	Message
1	21:19:42	SELECT distinct MaritalStatus FROM general_data LIMIT 0, 1000	3 row(s) returned

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

INPUT: SELECT * FROM general_data
WHERE TotalWorkingYears>2 AND YearsSinceLastPromotion<4;

OUTPUT:

31	Yes	Travel_Frequently	6	1
32	No	Travel_Frequently	5	0
32	No	Travel_Rarely	9	0
28	Yes	Travel_Rarely	5	0
29	No	Travel_Rarely	10	0
25	No	Non-Travel	6	1
28	No	Travel_Rarely	5	0
37	No	Travel_Rarely	7	0
21	No	Travel_Rarely	3	1
37	No	Non-Travel	15	0

SCREENSHOT:

10 • SELECT * FROM general_data
11 WHERE TotalWorkingYears>2 AND YearsSinceLastPromotion<4;

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

	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeID	Gender	
▶	31	Yes	Travel_Frequently	Research & Development	10	1	Life Sciences	1	2	Female	1
	32	No	Travel_Frequently	Research & Development	17	4	Other	1	3	Male	4
	32	No	Travel_Rarely	Research & Development	10	1	Medical	1	5	Male	1
	28	Yes	Travel_Rarely	Research & Development	11	2	Medical	1	7	Male	2
	29	No	Travel_Rarely	Research & Development	18	3	Life Sciences	1	8	Male	2
	25	No	Non-Travel	Research & Development	7	4	Medical	1	10	Female	4
	28	No	Travel_Rarely	Research & Development	1	3	Life Sciences	1	15	Male	1
	37	No	Travel_Rarely	Research & Development	1	2	Life Sciences	1	16	Male	1

general_data 18 x

Output

Action Output

#	Time	Action	Message
✓ 1	21:23:44	SELECT * FROM general_data WHERE TotalWorkingYears>2 AND YearsSinceLastPromotion<4;	1000 row(s) returned

11. List employees who have changed their job roles within the company (JobLevel and JobRole differ from their previous job).

INPUT: SELECT * FROM general_data
WHERE YearsAtCompany>0 AND YearsSinceLastPromotion>0;

OUTPUT:

31	Yes	6	3	5	1	4
38	No	13	5	8	7	5
46	No	28	5	7	7	7
31	No	10	2	9	7	8
25	No	6	2	6	1	5
45	No	21	2	20	4	10
36	No	16	2	15	10	11
55	No	37	2	36	4	13
47	Yes	10	4	10	9	9
21	No	3	3	3	1	0

SCREENSHOT:

The screenshot shows a SQL query execution interface. The query is: `SELECT * FROM general_data WHERE YearsAtCompany>0 AND YearsSinceLastPromotion>0;`. The results are displayed in a table with columns: Age, Attrition, BusinessTravel, Department, DistanceFromHome, Education, EducationField, EmployeeCount, EmployeeID, and Gender. The results show 10 rows of data.

Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeID	Gender
31	Yes	Travel_Frequently	Research & Development	10	1	Life Sciences	1	2	Female
38	No	Non-Travel	Research & Development	2	5	Life Sciences	1	4	Male
46	No	Travel_Rarely	Research & Development	8	3	Life Sciences	1	6	Female
31	No	Travel_Rarely	Research & Development	1	3	Life Sciences	1	9	Male
25	No	Non-Travel	Research & Development	7	4	Medical	1	10	Female
45	No	Travel_Rarely	Research & Development	17	2	Medical	1	11	Male
36	No	Travel_Rarely	Research & Development	28	1	Life Sciences	1	12	Male
55	No	Travel_Rarely	Research & Development	14	4	Life Sciences	1	13	Female
47	Yes	Travel_Frequently	Research & Development	10	4	Life Sciences	1	9	Male
21	No	Travel_Rarely	Research & Development	3	3	Life Sciences	1	1	Female

The interface also shows an "Action Output" section with a message: "1000 row(s) returned".

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

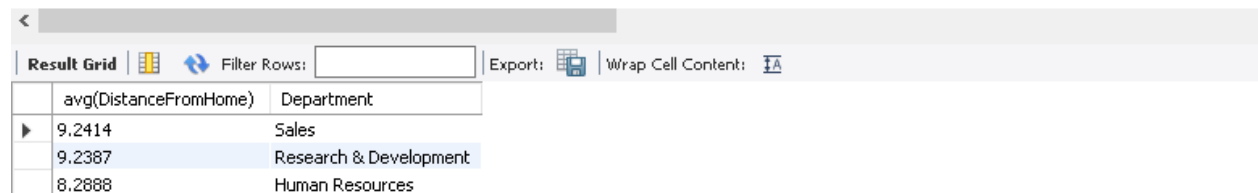
INPUT: SELECT avg(DistanceFromHome),Department
FROM general_data
GROUP BY Department;

OUTPUT:

9.2414	Sales
9.2387	Research & Development
8.2888	Human Resources

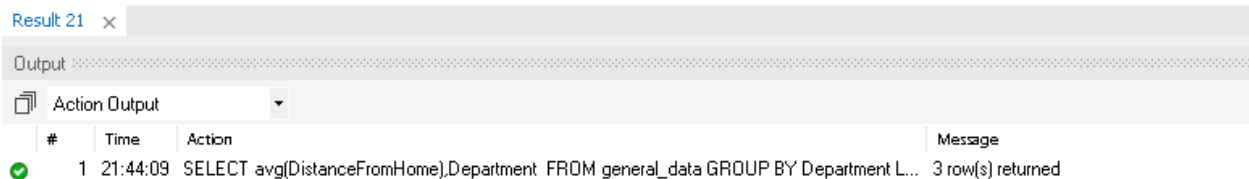
SCREENSHOT:

```
12 • SELECT avg(DistanceFromHome),Department
13 FROM general_data
14 GROUP BY Department;
```



The screenshot shows a SQL query editor with a query window and a results grid. The query is: `SELECT avg(DistanceFromHome),Department FROM general_data GROUP BY Department;`. The results grid displays the following data:

avg(DistanceFromHome)	Department
9.2414	Sales
9.2387	Research & Development
8.2888	Human Resources



The screenshot shows a SQL query editor with a query window and a results grid. The query is: `SELECT avg(DistanceFromHome),Department FROM general_data GROUP BY Department;`. The results grid displays the following data:

avg(DistanceFromHome)	Department
9.2414	Sales
9.2387	Research & Development
8.2888	Human Resources

13. Retrieve the top 5 employees with the highest MonthlyIncome.

INPUT: SELECT * FROM general_data
ORDER BY MonthlyIncome LIMIT 5;

OUTPUT:

58	Yes	Travel_Rarely	Sales	10090
58	Yes	Travel_Rarely	Sales	10090
58	Yes	Travel_Rarely	Sales	10090
36	No	Travel_Rarely	Research & Development	10510
36	No	Travel_Rarely	Research & Development	10510

SCREENSHOT:

The screenshot displays a SQL query execution environment. At the top, the query is entered in a text area:

```
13 • SELECT * FROM general_data
14 ORDER BY MonthlyIncome LIMIT 5;
```

Below the query area, a toolbar includes options like 'Result Grid', 'Filter Rows', 'Export', 'Wrap Cell Content', and 'Fetch rows'. The main area shows a table of results with 12 columns: Age, Attrition, BusinessTravel, Department, DistanceFromHome, Education, EducationField, EmployeeCount, EmployeeID, Gender, and JobLevel. The results are as follows:

	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeID	Gender	JobLevel
▶	58	Yes	Travel_Rarely	Sales	1	4	Technical Degree	1	3649	Male	2
	58	Yes	Travel_Rarely	Sales	1	4	Technical Degree	1	709	Male	2
	58	Yes	Travel_Rarely	Sales	1	4	Technical Degree	1	2179	Male	2
	36	No	Travel_Rarely	Research & Development	1	3	Life Sciences	1	2393	Female	2
	36	No	Travel_Rarely	Research & Development	1	3	Life Sciences	1	3863	Female	2

Below the table, there is a tab labeled 'general_data 22'. At the bottom, an 'Output' section shows the 'Action Output' table:

#	Time	Action	Message
✓ 1	21:43:14	SELECT * FROM general_data ORDER BY MonthlyIncome LIMIT 5	5 row(s) returned

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

INPUT:

```
SELECT(count(CASE when YearsSinceLastPromotion=1 then 1
end)/count(EmployeeID))*100
AS percentage from general_data;
```

OUTPUT: 24.2355

SCREENSHOT:

The screenshot displays a database query editor interface. At the top, the SQL query is entered in two lines:

```
1 SELECT(count(CASE when YearsSinceLastPromotion=1 then 1 end)/count(EmployeeID))*100
2 AS percentage from general_data;
```

Below the query editor, there is a toolbar with options like "Result Grid", "Filter Rows", "Export", and "Wrap Cell Content". The "Result Grid" is active, showing a single column named "percentage" with a value of 24.2355.

At the bottom, there is a section labeled "Output" with a dropdown menu set to "Action Output". Below this, a table shows the execution details:

#	Time	Action	Message
1	22:37:09	SELECT(count(CASE when YearsSinceLastPromotion=1 then 1 end)/count(EmployeeID))*100 ...	1 row(s) returned

15. List the employees with the highest and lowest EnvironmentSatisfaction.

INPUT: SELECT * FROM employee_survey_data
WHERE EnvironmentSatisfaction=4 OR EnvironmentSatisfaction=1;

OUTPUT:

4	4	4	3
5	4	1	3
7	1	3	1
8	1	2	3
13	4	1	3
14	1	2	2
15	4	4	2
17	4	3	4
18	1	4	3
20	1	1	3

SCREENSHOT:

The screenshot displays a SQL query execution environment. At the top, the query is entered in a text area:

```
15 • SELECT * FROM employee_survey_data
16 WHERE EnvironmentSatisfaction=4 OR EnvironmentSatisfaction=1;
```

Below the query area, a toolbar includes options like 'Filter Rows', 'Export', 'Wrap Cell Content', and 'Fetch rows'. The main area shows a 'Result Grid' with the following data:

EmployeeID	EnvironmentSatisfaction	JobSatisfaction	WorkLifeBalance
4	4	4	3
5	4	1	3
7	1	3	1
8	1	2	3
13	4	1	3
14	1	2	2
15	4	4	2
17	4	3	4
18	1	4	3
20	1	1	3

Below the result grid, there is an 'Output' section with a dropdown menu set to 'Action Output'. The output log shows the following entry:

#	Time	Action	Message
✓ 1	21:59:50	SELECT * FROM employee_survey_data WHERE EnvironmentSatisfaction=4 OR Environmen...	1000 row(s) returned

16. Find the employees who have the same JobRole and MaritalStatus.

INPUT:

```
SELECT * FROM general_data
WHERE (JobRole, MaritalStatus)
IN (SELECT JobRole, MaritalStatus FROM general_data
GROUP BY JobRole, MaritalStatus
HAVING count(EmployeeID)>1);
```

OUTPUT:

51	No	Healthcare Representative	Married
31	Yes	Research Scientist	Single
32	No	Sales Executive	Married
38	No	Human Resources	Married
32	No	Sales Executive	Single
46	No	Research Director	Married
28	Yes	Sales Executive	Single
29	No	Sales Executive	Married
31	No	Laboratory Technician	Married
25	No	Laboratory Technician	Divorced

SCREENSHOT:

```

3 • SELECT * FROM general_data
4 WHERE (JobRole, MaritalStatus)
5 IN (SELECT JobRole, MaritalStatus FROM general_data
6 GROUP BY JobRole, MaritalStatus
7 HAVING count(EmployeeID)>1);

```

Result Grid

	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeID	Gender
51	No	Travel_Rarely	Sales		6	2	Life Sciences	1	1	Female
31	Yes	Travel_Frequently	Research & Development		10	1	Life Sciences	1	2	Female
32	No	Travel_Frequently	Research & Development		17	4	Other	1	3	Male
38	No	Non-Travel	Research & Development		2	5	Life Sciences	1	4	Male
32	No	Travel_Rarely	Research & Development		10	1	Medical	1	5	Male
46	No	Travel_Rarely	Research & Development		8	3	Life Sciences	1	6	Female
28	Yes	Travel_Rarely	Research & Development		11	2	Medical	1	7	Male
20	No	Travel_Rarely	Research & Development		18	2	Life Sciences	1	8	Male

general_data 32 x

Output

Action Output

#	Time	Action	Message
1	22:44:20	SELECT * FROM general_data WHERE (JobRole, MaritalStatus) IN (SELECT JobRole, MaritalStatus FROM general_data GROUP BY JobRole, MaritalStatus HAVING count(EmployeeID)>1);	1000 row(s) returned

17. List the employees with the highest TotalWorkingYears who also have a PerformanceRating of 4.

INPUT:

```

SELECT * FROM general_data
INNER JOIN manager_survey_data
ON general_data.EmployeeID=manager_survey_data.EmployeeID
WHERE manager_survey_data.PerformanceRating=4
ORDER BY TotalWorkingYears DESC LIMIT 1;

```

OUTPUT:

53	No...	...35	2	9	8	8	225	6	3	4
----	-------	-------	---	---	---	---	-----	---	---	---

SCREENSHOT:

```

1 • SELECT * FROM general_data
2 INNER JOIN manager_survey_data
3 ON general_data.EmployeeID=manager_survey_data.EmployeeID
4 WHERE manager_survey_data.PerformanceRating=4
5 ORDER BY TotalWorkingYears DESC LIMIT 1;

```

Result Grid										
	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeID	Gender
	53	No	Travel_Rarely	Research & Development	2	4	Technical Degree	1	2256	Male

Output			
Action Output			
#	Time	Action	Message
✓ 1	23:00:47	SELECT * FROM general_data INNER JOIN manager_survey_data ON general_data.EmployeeID=manager_survey_data.EmployeeID WHERE manager_survey_data.PerformanceRating=4 ORDER BY TotalWorkingYears DESC LIMIT 1;	1 row(s) returned

18. Calculate the average Age and JobSatisfaction for each BusinessTravel type.

INPUT:

```

SELECT avg(Age),avg(JobSatisfaction),BusinessTravel FROM general_data
INNER JOIN employee_survey_data
ON general_data.EmployeeID=employee_survey_data.EmployeeID
GROUP BY BusinessTravel;

```

OUTPUT:

37.1016	2.704	Travel_Rarely
36.4739	2.7844	Travel_Frequently
36.6116	2.7915	Non-Travel

SCREENSHOT:

```
1 • SELECT avg(Age), avg(JobSatisfaction), BusinessTravel FROM general_data
2 INNER JOIN employee_survey_data
3 ON general_data.EmployeeID=employee_survey_data.EmployeeID
4 GROUP BY BusinessTravel;
```

Result Grid

	avg(Age)	avg(JobSatisfaction)	BusinessTravel
▶	37.1016	2.7040	Travel_Rarely
	36.4739	2.7844	Travel_Frequently
	36.6116	2.7915	Non-Travel

Result 41 x

Output

Action Output

#	Time	Action	Message
✓ 1	23:09:40	SELECT avg(Age), avg(JobSatisfaction), BusinessTravel FROM general_data INNER JOIN emp...	3 row(s) returned

19. Retrieve the most common EducationField among employees.

INPUT:

```
SELECT EducationField, count(*) AS FIELD_COUNT
FROM general_data
GROUP BY EducationField
ORDER BY FIELD_COUNT DESC LIMIT 1;
```

OUTPUT:

Life Sciences	1806
---------------	------

SCREENSHOT:

```

1 • SELECT EducationField, count(*) AS FIELD_COUNT
2 FROM general_data
3 GROUP BY EducationField
4 ORDER BY FIELD_COUNT DESC LIMIT 1;
5

```

Result Grid

EducationField	FIELD_COUNT
Life Sciences	1806

Result 42 x

Output

Action Output

#	Time	Action	Message
✓ 1	23:15:38	SELECT EducationField, count(*) AS FIELD_COUNT FROM general_data GROUP BY Educat...	1 row(s) returned

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

INPUT: SELECT * FROM general_data
 WHERE YearsSinceLastPromotion=0
 ORDER BY YearsAtCompany DESC LIMIT 1;

OUTPUT:

51	No	33	2	33	0	10
----	----	----	---	----	---	----

SCREENSHOT:

```
1 SELECT * FROM general_data
2 WHERE YearsSinceLastPromotion=0
3 ORDER BY YearsAtCompany DESC LIMIT 1;
```

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

Fetch rows:

	Age	Attrition	BusinessTravel	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeID	Gender	JobLevel	JobRole
▶	51	No	Travel_Rarely	Sales	18	2	Life Sciences	1	2175	Male	4	Manufacturing

general_data 29

x

Output

Action Output

#	Time	Action	Message
✓ 1	22:25:54	SELECT * FROM general_data WHERE YearsSinceLastPromotion=0 ORDER BY YearsAtCom...	1 row(s) returned