JustIT Data Skills Workshop Week 5

Project: Analyzing Employee Data Using MySQL and Power BI

By Riyad Hasan

Questions

1. What is the total number of employees in the company?

To answer this question, an SQL query was used, it reads as follows:

```
-- total employees
select
count(distinct(employees.emp_no)) AS TotalEmployees
from employees
;
```

Distinct is used to ensure that repeated employee numbers i.e. the same employee are not included in the count. Using this query returns that the total number of employees is **300,024.**

A card visual was created on Power BI displaying this result.

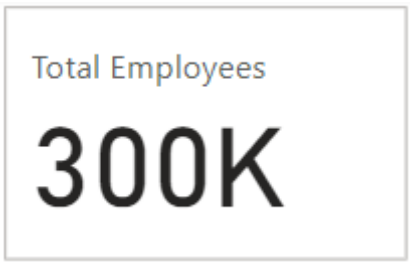

Figure 1. Card Visual of Total Number of Employees

2. What is the average salary by department?

To answer this question, joins were used to relate several different tables including the salaries, employees, demt_emp and departments table. The query is written below:

```
-- avg salary by dept
select departments.dept_no
departments.dept_name,
avg(salaries.salary) AS AvgSalary
from salaries
inner join employees
on salaries.emp_no = employees.emp_no
inner join dept_emp
on employees.emp_no = dept_emp.emp_no
inner join departments
on dept_emp.dept_no = departments.dept_no
group by departments.dept_name
order by AvgSalary
;
```

The results of this query have been tabulated below:

dept_no	dept_name	AvgSalary
d007	Sales	80667.6058
d001	Marketing	71913.2000
d002	Finance	70489.3649
d008	Research	59665.1817
d004	Production	59605.4825
d005	Development	59478.9012

d009	Customer Service	58770.3665
d006	Quality Management	57251.2719
d003	Human Resources	55574.8794

This shows that the Sales department has the highest average salary, whilst Human Resources department has the lowest average salary.

A bar chart displaying this was created on Power BI.

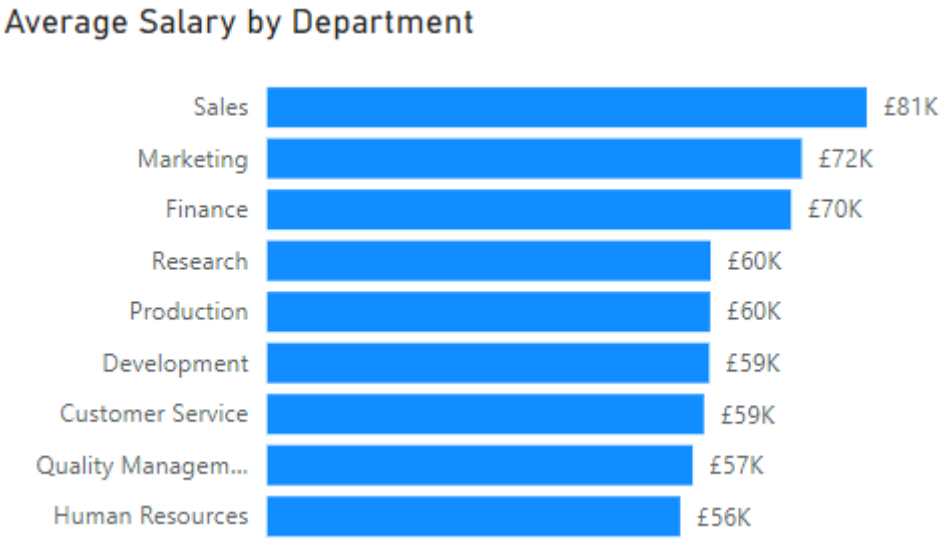

Figure 2. Bar Chart showing Average Salary per Department

3. Who are the top 10 highest paid employees?

The following query was used to answer this question:

```
-- top 10 highest paid employees
select employees.emp_no,
       employees.first_name,
       employees.last_name,
       salaries.salary
from employees
inner join salaries
on employees.emp_no = salaries.emp_no
order by salaries.salary desc
limit 10
;
```

The results of the above query have been tabulated below.

emp_no	first_name	Last_name	salary
43624	Tokuyasu	Pesch	158220
43624	Tokuyasu	Pesch	157821
254466	Honesty	Mukaidono	156286
47978	Xiahua	Whitcomb	155709
253939	Sanjai	Luders	155513
109334	Tsutomu	Alameldin	155377
109334	Tsutomu	Alameldin	155190
109334	Tsutomu	Alameldin	154888

109334	Tsutomu	Alameldin	154885
80823	Willard	Baca	154459

4. How many employees are there in each department?

The following query was used:

```
-- employees count by dept
select departments.dept_name,
count(distinct(employees.emp_no)) AS TotalEmployees
from employees
inner join dept_emp
on employees.emp_no = dept_emp.emp_no
inner join departments
on dept_emp.dept_no = departments.dept_no
group by departments.dept_name
order by TotalEmployees desc
;
```

The following table was generated.

dept_name	TotalEmployees
Development	85707
Production	73485
Sales	52245
Customer Service	23580
Research	21126
Marketing	20211
Quality Management	20117
Human Resources	17786
Finance	17346

This shows Development department has the most employees whilst the Finance department has the least.

A pie chart to show this distribution was created.

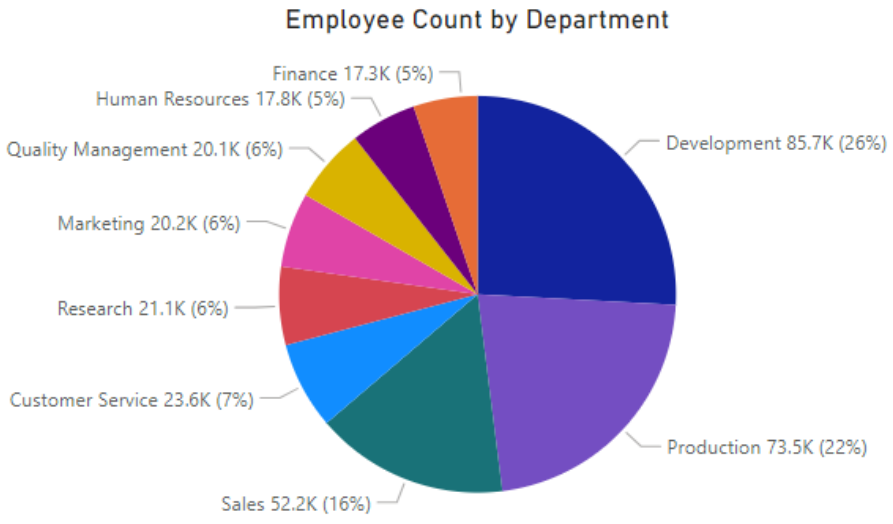

Figure 3. Pie chart of Employee Count by Department

5. What is the trend of average salaries over the years?

To answer this question, a yearly basis approach was taken. The query below was used:

```
-- salary trend over time (yearly trend)
select YEAR(to_date) AS year,
avg(salary) AS AvgSalary
from salaries
group by year
having year != 9999
order by year asc
;
```

In Line 1, the YEAR() function is used to extract only the year from the full date since a yearly basis approach is being used. Line 5 is used to remove open-ended salaries data entries from the query, since many of the entries in the salaries table are not complete and use 9999 as the year to convey the date of the salary is not known. The results are ordered by years ascending so a trend over time can be observed.

The following table was returned.

Year	AvgSalary
1985	51542.7753
1986	53182.9938
1987	54104.2295
1988	54967.9753
1989	55871.6536
1990	56849.8019
1991	57844.4753
1992	58796.7907
1993	59778.0252
1994	60756.9129
1995	61729.9795
1996	62687.4633
1997	63615.9766
1998	64567.3876
1999	65533.1352
2000	66549.9994
2001	68597.4674
2002	70558.5668

Creating a rough line chart in excel shows the trend:

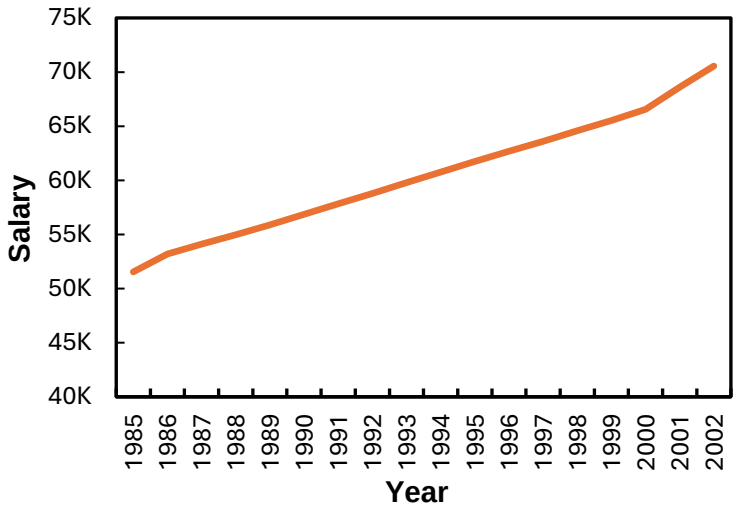

Figure 4. Yearly trend of Salary

This shows that the salary has steadily increased over time.

6. What is the gender distribution within each department?

To calculate this, a longer query was used:

```
-- gender distribution by department
select
departments.dept_name,
sum(case when employees.gender = "M" then 1 else 0 END) as Males,
sum(case when employees.gender = "F" then 1 else 0 END) as Females,
count(*) AS Total
from employees
inner join dept_emp
on employees.emp_no = dept_emp.emp_no
inner join departments
on dept_emp.dept_no = departments.dept_no
group by departments.dept_name
order by Total desc
;
```

Line 3 uses the case when expression to sum all instances when there is a row in the gender column of the employees table where the gender is “M” i.e. Male. Line 4 does the same when gender is “F” i.e. female. The query groups by department name so the gender distribution of employees for the department is outputted, shown below.

dept_name	Males	Females	Total
Development	51449	34258	85707
Production	43936	29549	73485
Sales	31391	20854	52245
Customer Service	14132	9448	23580
Research	12687	8439	21126
Marketing	12174	8037	20211
Quality Management	12039	8078	20117
Human Resources	10711	7075	17786
Finance	10331	7015	17346

Two charts were created for this. One is a stacked column chart showing the gender distribution and total count of employees in each department. The other is a 100% stacked column chart showing the percentage gender distribution of employees in each department.

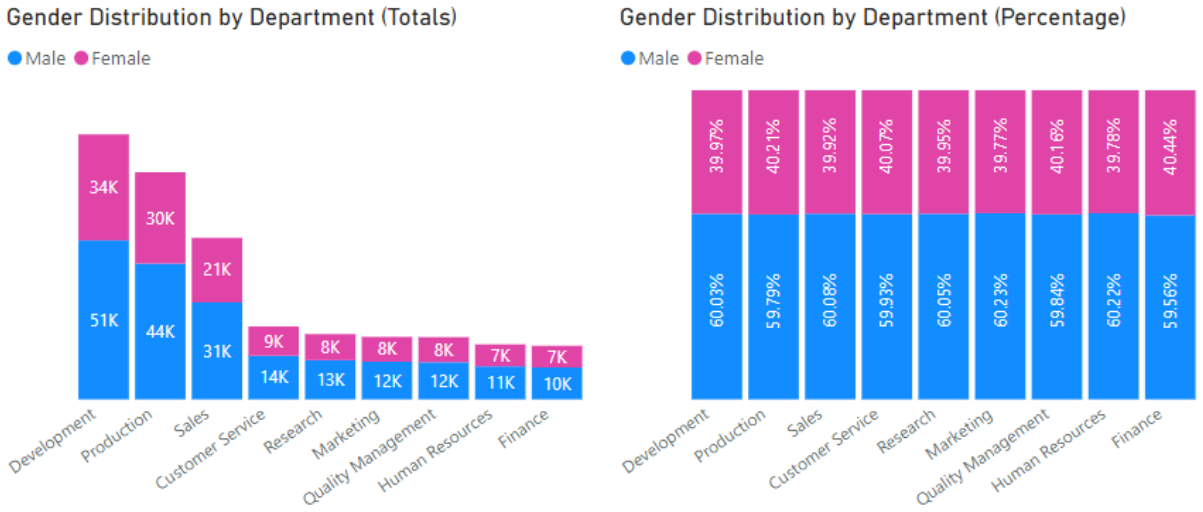

Figure 5. Gender distribution of employees in each department

The chart on the left shows the gender distribution along with the totals, so we can see the employee counts for each department and their gender breakdown. Whereas the chart on the right only shows the gender distribution so insights pertaining to that are more apparent. For example, we can clearly see that the gender distribution across all departments is 60% Male and 40% Female. This means the gender distribution of the company is 60:40 male: female.

Advanced Analytical Questions

1. Employee Retention Analysis

The following query was used:

```
select
departments.dept_name,
avg(ABS(DATEDIFF(to_date, from_date))) AS Tenure
from dept_emp
inner join departments
on dept_emp.dept_no = departments.dept_no
where to_date != '9999-01-01'
group by departments.dept_name
order by Tenure desc
;
```

DATEDIFF() in Line 3 returns the difference between two dates in the form of days. So here, it finds the difference between the date an employee began their role and the date they left and outputs this in days. ABS() is used to ensure absolute values are used in case the difference is outputted as a negative value. In Line 7 we use "!=" to exclude the date '9999-01-01' from the query because this is used as a dummy date for unknown data or for employees who are currently working in the role. We do not want to consider presently working employees as their tenure is still ongoing. If we do not exclude this, the results will be skewed.

Using this query, the following table was outputted.

dept_name	Tenure
Finance	1697.6763
Development	1693.9246
Sales	1684.3591
Human Resources	1679.3472
Production	1646.9722
Quality Management	1591.6703
Research	1584.6174
Marketing	1559.7880
Customer Service	1541.8701

Using this data, a bar chart is created in Power BI.

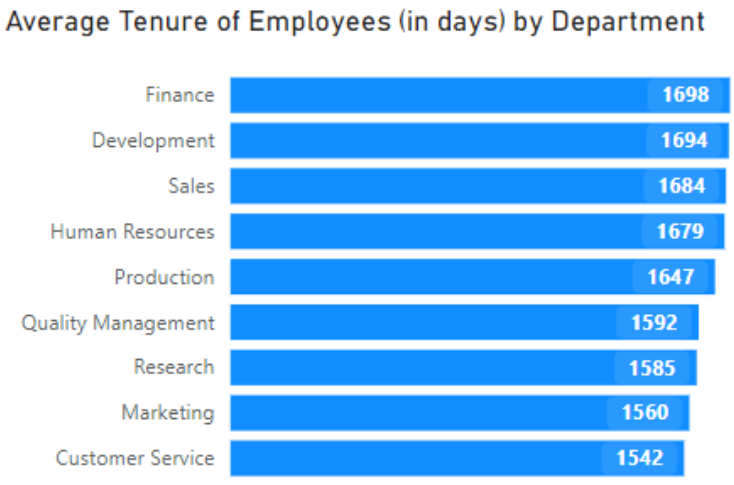

Figure 6. Average Tenure of Employees by Department

Finance department has the longest average tenure whilst customer service has the shortest. However, there is not a wide difference for average tenure of employees across departments.

2. Salary Progression Analysis

The dataset is insufficient to answer this question as it does not contain any information about promotions. Thus the question cannot be answered.

3. Gender Pay Gap Analysis

The following query was used.

```
select
departments.dept_name,
avg(case when employees.gender = "M" then salary END) as MaleAvgSalary,
avg(case when employees.gender = "F" then salary END) as FemaleAvgSalary,
avg(salary) as AvgSalary
from employees
inner join salaries
on employees.emp_no = salaries.emp_no
inner join dept_emp
on salaries.emp_no = dept_emp.emp_no
inner join departments
on dept_emp.dept_no = departments.dept_no
group by departments.dept_name
;
```

Lines 3 and 4 use the case when expression to find the average of all entries that meet a specific condition, in this case when there is a row in the gender column of the employees table where the gender is “M” or “F” i.e. Male or Female.

The following table was generated.

dept_name	MaleAvgSalary	FemaleAvgSalary	AvgSalary
Development	59507.6407	59435.6436	59478.9012
Sales	80736.6783	80563.8577	80667.6058
Production	59676.2339	59500.2896	59605.4825
Human Resources	55462.1302	55744.9599	55574.8794
Research	59656.4158	59678.3021	59665.1817
Quality Management	57214.7315	57305.1893	57251.2719
Customer Service	58682.6523	58901.5697	58770.3665
Marketing	72087.8209	71643.4196	71913.2000
Finance	70456.5559	70537.7665	70489.3649

Looking at this data, we can see that male and female average salaries are very close together for each department. There are some departments where the male average salary is slightly higher whilst there are some departments where the female average salary is higher. Thus, it seems there is no gender pay gap. To visualise this a clustered column chart was created on Power BI.

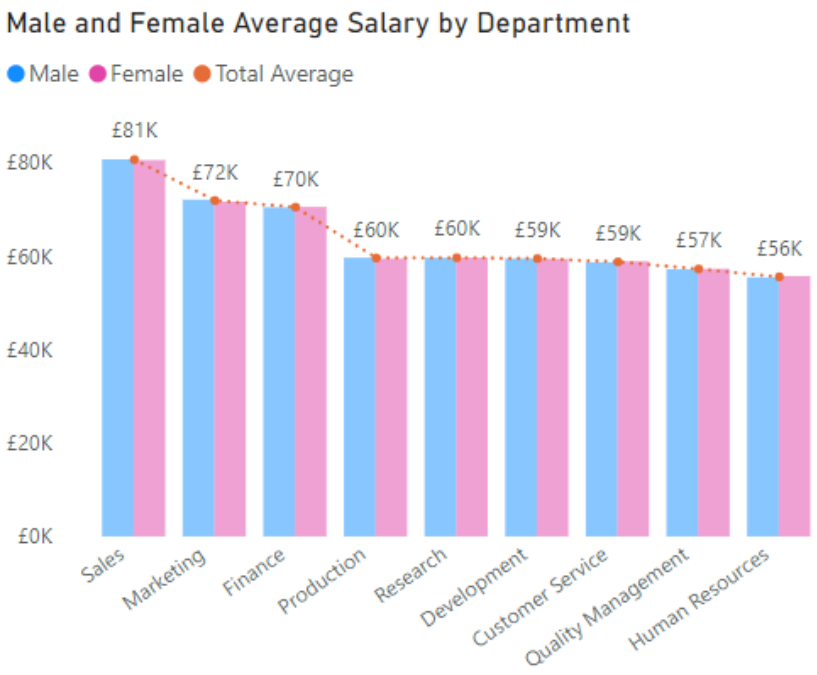

Figure 7. Male and Female Average Salary by Department.

Based on this chart, we can see that male and female average salaries across each department are almost identical and match the average salary for each department. Therefore, we can say there is no gender pay gap in any department in this company.

4. Promotion Rate Analysis

The dataset is insufficient to answer this question as it does not contain any information about promotions. Thus the question cannot be answered.

5. Managerial Effectiveness Analysis

The dataset is insufficient to answer this question as it does not contain any information about the specific manager of each employee; it only tells us the department the employee works in and each department can have multiple managers. Thus the question cannot be answered.

6. Hiring Trends Analysis

The following query was used.

```
select YEAR(dept_emp.from_date) AS year,
sum(case when dept_no = "d001" then 1 END) as Marketing,
sum(case when dept_no = "d002" then 1 END) as Finance,
sum(case when dept_no = "d003" then 1 END) as HumanResources,
sum(case when dept_no = "d004" then 1 END) as Production,
sum(case when dept_no = "d005" then 1 END) as Development,
sum(case when dept_no = "d006" then 1 END) as QualityManagement,
sum(case when dept_no = "d007" then 1 END) as Sales,
sum(case when dept_no = "d008" then 1 END) as Research,
sum(case when dept_no = "d009" then 1 END) as CustomerService
from employees
inner join dept_emp
on dept_emp.emp_no = employees.emp_no
group by year
having year != 9999
order by year asc;
```

Line 1 selects the year from each start date of the role for each employee, since we want to investigate the trend over years. Lines 2-10 use case when functions to sum each instance of a new hire for each department. This will create columns for the number of hires for each department. The data is grouped by year so each year will be a new row. This allows us to find the hiring trends of each department over time.

The following table is outputted (dept_no is used for column headings so the table fits the page. Refer to the query above to see the department name each dept_no corresponds to).

Year	d001	d002	d003	d004	d005	d006	d007	d008	d009
1985	1040	1041	1001	4127	5092	977	3090	1039	973
1986	1100	1097	1117	4529	5456	1109	3351	1179	1216
1987	1103	1103	1178	4524	5584	1202	3362	1154	1224
1988	1174	1158	1147	4623	5513	1218	3416	1200	1287
1989	1174	1185	1143	4556	5754	1216	3459	1255	1301
1990	1217	1150	1152	4697	5608	1228	3460	1231	1296
1991	1257	1086	1185	4629	5541	1258	3342	1318	1368
1992	1307	1119	1139	4743	5827	1211	3492	1318	1392
1993	1369	1128	1154	4714	5614	1284	3437	1344	1458
1994	1307	1102	1255	4650	5623	1265	3558	1399	1459
1995	1238	1163	1205	4899	5749	1293	3487	1351	1523
1996	1388	1210	1231	4974	5832	1432	3372	1449	1625
1997	1394	1177	1181	5167	5768	1387	3481	1450	1767
1998	1480	1140	1161	5102	5739	1533	3477	1508	1793
1999	1543	1146	1199	5116	5740	1483	3682	1608	1835
2000	490	202	202	1209	810	481	492	581	883
2001	416	86	82	802	310	335	194	481	765
2002	214	53	54	424	147	205	93	261	415

Using Power BI, a stacked area chart is created using this data.

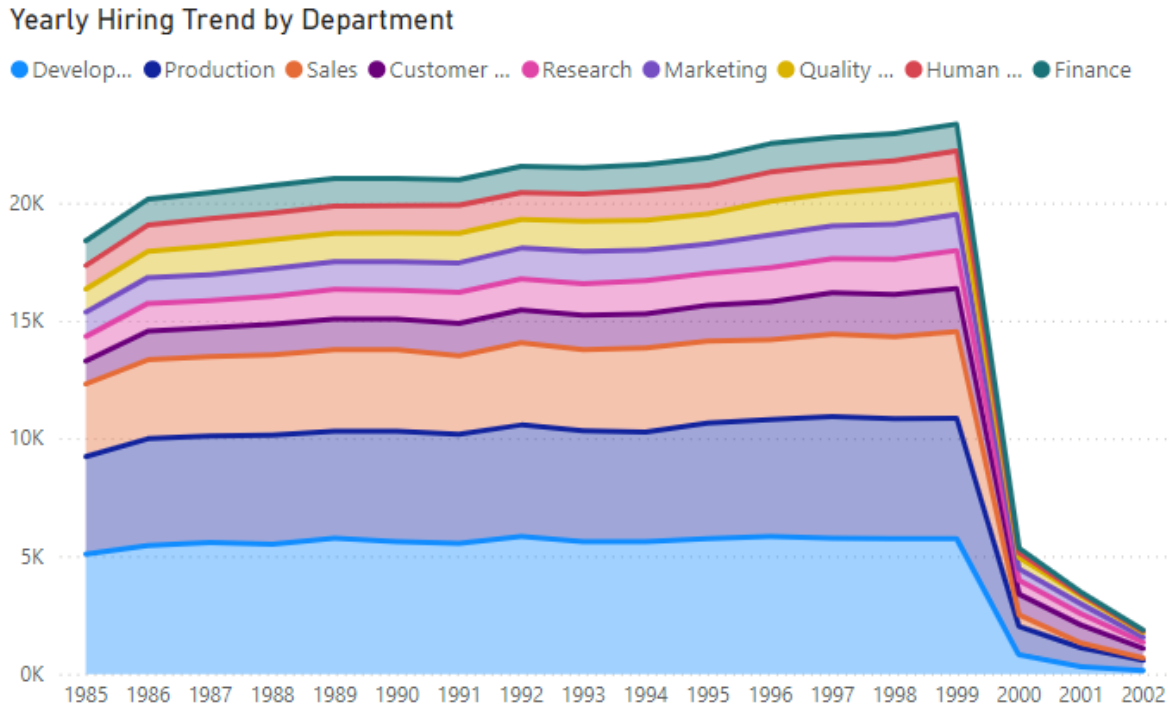

Figure 8. Hiring trends over time by Department

The chart shows that from 1985-1999, all departments had a slow rise in their hiring numbers, with some short periods of minor hikes or slumps. The largest departments, Development and Production, had the most stagnant hiring trend whilst the smaller departments, like Finance and Human Resources, had slightly more hiring growth year on year. In 2000, there was a major drop in hiring numbers for all departments, suggesting the company may have had a hiring freeze. This trend continues for the remainder of the years (2001 and 2002) where the hiring numbers continued to fall.

7. Employee Age Distribution

Before creating a query to find the age distribution, we create a generated column in the employee table to find the age of each employee, called emp_age.

```
ALTER TABLE employees
ADD emp_age INT
      AS (TIMESTAMPDIFF(YEAR, employees.birth_date, "2024-08-10"))
;
```

ALTER TABLE indicates which table we want to manipulate, in this case we want to add something to the employees table. ADD indicates we want to add a new column, called emp_age and set as the INT (integer) data type. AS indicates we want to set the values for each row in this column with the following. In this case, we want to find the age in years of each employee on the date 10th August 2024. We use TIMESTAMPDIFF() to determine the age, it has three arguments; The time unit we want the age outputted in, in this case years. The start date, in this case the birth date of each employee. And the end date, in this case the date 10th August 2024. This successfully creates a generated column called emp_age.

Now a query can be made, as follows.

```
select
departments.dept_name,
count(case when emp_age between 59 and 60 THEN 1 END) AS "59-60",
count(case when emp_age between 61 and 62 THEN 1 END) AS "61-62",
count(case when emp_age between 63 and 64 THEN 1 END) AS "63-64",
count(case when emp_age between 65 and 66 THEN 1 END) AS "65-66",
count(case when emp_age between 67 and 68 THEN 1 END) AS "67-68",
count(case when emp_age between 69 and 70 THEN 1 END) AS "69-70",
count(case when emp_age >= 71 THEN 1 END) AS "71+",
count(*) AS total
from employees
inner join dept_emp
on dept_emp.emp_no = employees.emp_no
inner join departments
on dept_emp.dept_no = departments.dept_no
group by departments.dept_name
;
```

Since we want to create a histogram, we use age bins. Using max() and min() functions for emp_age, we discover the max age is 72 and min age is 59, so these are the max and min values for each age bin. We decide to use 7 age bins, each bin having 2 ages i.e. our age bins are: 59-60, 61-62 … 69-70 and 71+. Lines 3-7 are used to find the counts for each age bin using the case when function for the ages that belong in the age bin. For example. Line 3 queries MySQL to count each row where the employees age is between (and including) 59 and 60. This is performed for the remaining age bins. The data is also grouped by department. The following table was outputted:

dept_name	59-60	61-62	63-64	65-66	67-68	69-70	71+
Customer Service	2731	3622	3566	3551	3645	3758	2707
Development	9693	13170	13439	13251	13146	13164	9844
Finance	1996	2616	2686	2645	2672	2681	2050
Human Resources	2088	2622	2725	2752	2639	2780	2180
Marketing	2259	3048	3128	3216	3120	3122	2318
Production	8169	11414	11333	11320	11207	11298	8744
Quality Management	2203	3122	3092	3095	3050	3184	2371
Research	2542	3163	3281	3183	3147	3251	2559
Sales	5863	8085	7993	8125	8082	8019	6078

Using Excel, stacked column charts were created to emulate a histogram. The x-axis counted each age bin whilst the y-axis contained the counts of employees for each age bin, effectively emulating a histogram. 2 charts were created. One shows the total age distribution whilst the other is divided by department.

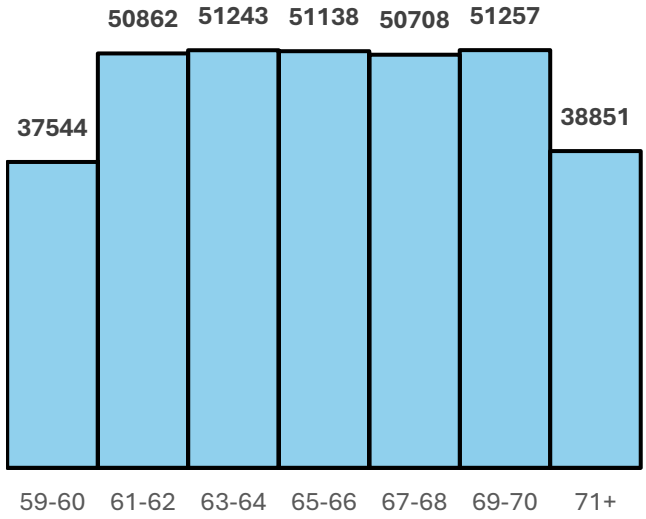

Figure 9. Histogram of Age Distribution of Employees

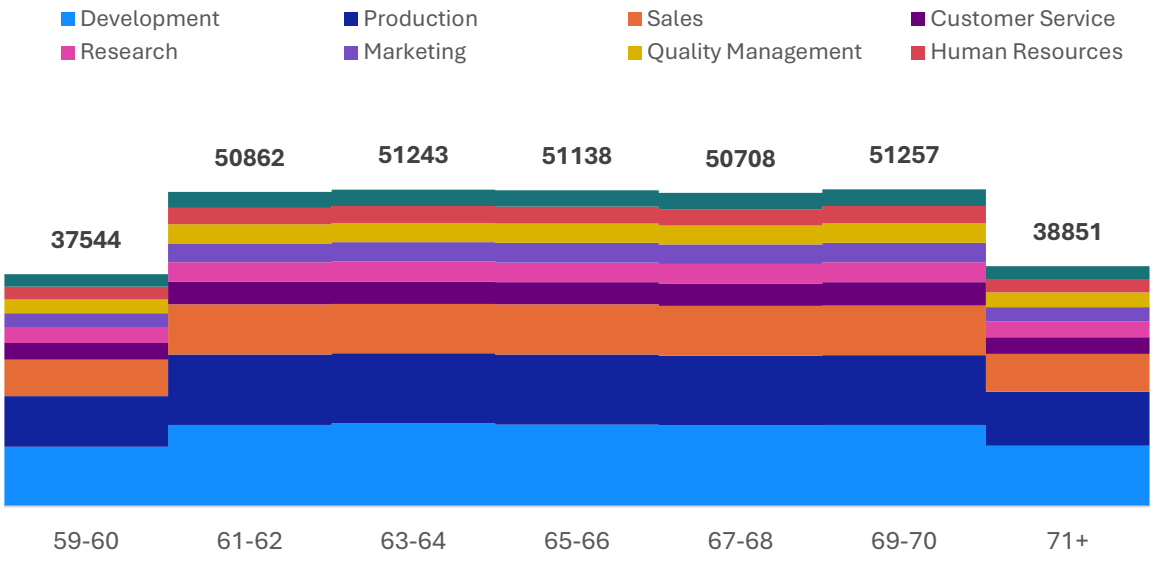

Figure 10. Histogram of Employee Age Distribution by Department

The histograms shows that the employee age frequency is most dense in the five inside age bins and least dense in the outside age bins (59-60 and 71+). However, all five interior bins have a close frequency. The data has a low amount of skew so we can say it is symmetrical data. The distribution only has one peak so it possibly follows a normal distribution. The distribution is platykurtic (negative kurtosis) i.e. the distribution curve is flatter than a normal distribution where the values are less concentrated around the middle bin. The histogram that is separated by department shows that all departments follow the same distribution.

8. Departmental Performance Analysis

The following query is used.

```
select YEAR(salaries.from_date) AS year,
ROUND(avg(case when dept_no = "d001" then salary END),2) as Marketing,
ROUND(avg(case when dept_no = "d002" then salary END),2) as Finance,
ROUND(avg(case when dept_no = "d003" then salary END),2) as HumanResources,
ROUND(avg(case when dept_no = "d004" then salary END),2) as Production,
ROUND(avg(case when dept_no = "d005" then salary END),2) as Development,
ROUND(avg(case when dept_no = "d006" then salary END),2) as
QualityManagement,
ROUND(avg(case when dept_no = "d007" then salary END),2) as Sales,
ROUND(avg(case when dept_no = "d008" then salary END),2) as Research,
ROUND(avg(case when dept_no = "d009" then salary END),2) as CustomerService
from salaries
inner join dept_emp
on dept_emp.emp_no = salaries.emp_no
group by year
having year != 9999
order by year asc
;
```

The case when function is used again to find the average salary of entries that meet a condition, in this case when the row is a specific dept_no. This function is nested within a ROUND() function so the outputted average is to 2 decimal places since it's a currency.

The following table was generated (dept_no is used for column headings so the table fits the page. Refer to the query above to see the department name each dept_no corresponds to).

Year	d001	d002	d003	d004	d005	d006	d007	d008	d009
1985	61614.57	60285.39	44956.80	48924.27	48706.24	46644.21	69975.04	49098.05	48195.14
1986	62379.13	61078.81	45990.08	49865.65	49676.34	47458.51	70909.17	49992.76	49081.52
1987	63151.05	61838.28	46780.81	50770.94	50609.80	48234.48	71867.92	50950.01	49940.20
1988	64027.49	62671.56	47598.95	51700.66	51596.05	49249.16	72796.53	51630.08	50806.60
1989	65069.63	63667.71	48591.53	52702.98	52546.74	50238.07	73700.45	52711.22	51645.64
1990	66057.55	64651.19	49552.67	53633.87	53530.38	51261.53	74701.31	53752.32	52641.50
1991	66959.96	65581.09	50491.18	54612.60	54501.91	52236.98	75700.20	54677.77	53600.37
1992	67866.46	66588.74	51498.36	55532.41	55420.83	53266.58	76637.19	55567.61	54635.97
1993	68795.01	67511.21	52463.74	56580.33	56404.96	54256.32	77594.13	56529.05	55641.58
1994	69758.67	68511.62	53435.23	57552.81	57375.95	55189.46	78536.81	57434.77	56635.63
1995	70793.81	69332.83	54369.71	58482.82	58334.30	56202.43	79470.75	58397.94	57662.33
1996	71730.81	70268.24	55283.82	59447.48	59284.03	57116.78	80474.60	59362.89	58636.14
1997	72687.65	71193.91	56293.54	60375.54	60230.54	58059.90	81447.48	60421.26	59558.86
1998	73642.32	72170.06	57338.39	61338.96	61207.56	58960.41	82452.91	61424.09	60498.42
1999	74498.73	73163.24	58358.06	62336.36	62194.85	59930.48	83352.61	62427.98	61506.60
2000	76563.73	75184.64	60394.79	64372.65	64228.09	61949.64	85379.02	64439.27	63530.60
2001	78686.35	77346.87	62523.60	66512.96	66363.76	64088.74	87516.13	66567.50	65659.40
2002	80512.40	79257.90	64486.00	68434.13	68335.69	66012.17	89480.37	68792.22	67657.34

Using this data, a line chart was created on Power BI.

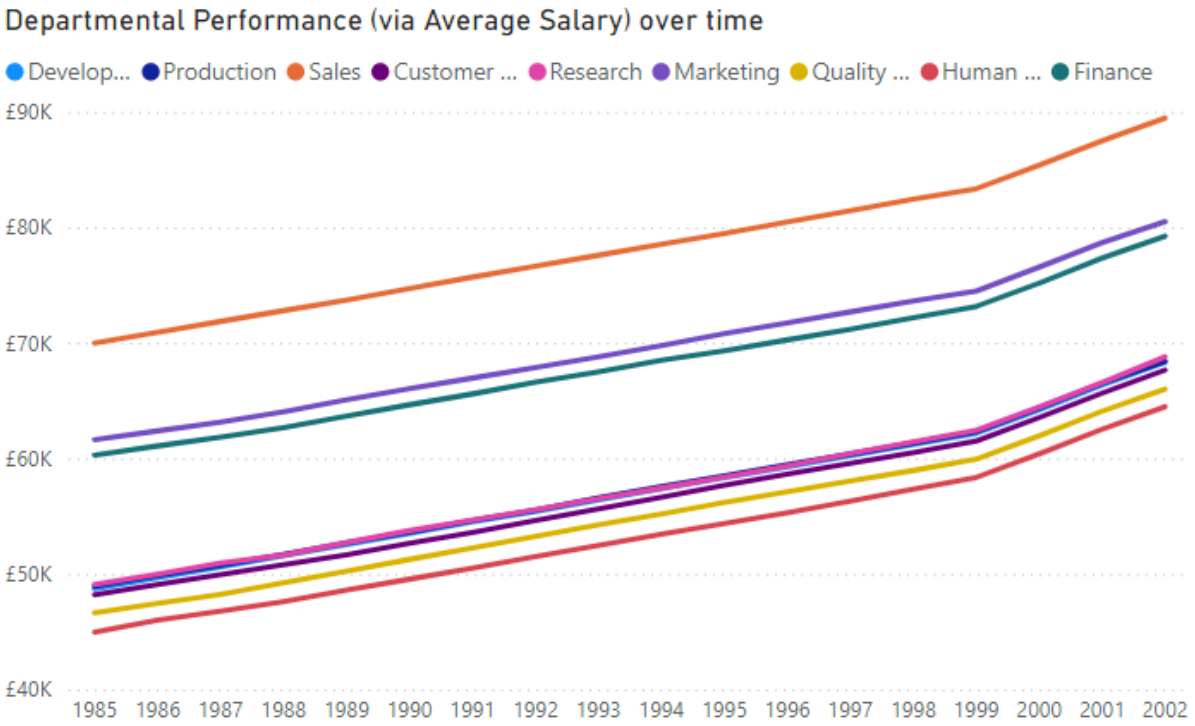

Figure 11. Hiring trends over time, by department.

This chart shows that average salary for all departments grew steadily over time. We can see a steady growth for all departments from 1985-1999, where there is an increase in growth rate. All departments share the same growth rate pattern. As such, we can conclude that departmental performance, measured via average salary, became better over time for all departments. Interestingly, this spike in average salary growth coincides with a sharp decline in hiring, which perhaps suggests the company downsized in its staff numbers and paid its remaining employees more.

9. Employee Turnover Analysis

The turnover rate can be calculated with the following formula.

$$\text{TurnoverRate} = \frac{\text{Employees who have left}}{\frac{\text{Employees at start date} + \text{Employees at end date}}{2}} \times 100$$

The following query is used.

```
-- finding the start date
select
MIN(from_date)
from dept_emp
;
-- full query
Select
departments.dept_name,
(sum(case when dept_emp.to_date != "9999-01-01" then 1 END) /
    (sum(case when dept_emp.to_date = "9999-01-01" then 1 END) +
    sum(case when dept_emp.from_date = "1985-01-01" then 1 END)
    ) / 2
) * 100 AS TurnoverRate
from dept_emp
inner join departments
```

```
on dept_emp.dept_no = departments.dept_no
group by departments.dept_name
order by TurnoverRate asc
;
```

The formula begins on Line 3 until 7, where several case when statements are used. Employees who left can be found by summing every instance when to_date (date employee worked until) is not 9999-01-01, since this date is used as a dummy to represent employees who are currently working in the role. So by counting all employees who don't have this date, we can count all the employees who have left the company. Conversely, employees at end date i.e. employers currently working at the company can be found by summing every instance when to_date is 9999-01-01. Employees at start date can be found by first finding the minimum from_date (i.e. start date) using the MIN() function. This was returned as 1985-01-01. Thus, by summing every instance when from_date is 1985-01-01 we can count all the employees at the start date.

This query generates the table below.

dept_name	TurnoverRate
Customer Service	17.10586225
Marketing	18.08596640
Research	18.40758965
Production	18.92974390
Human Resources	18.94720520
Quality Management	19.14827795
Sales	19.28810140
Finance	19.73388000
Development	19.80956880

A bar chart displaying this was created on Power BI.

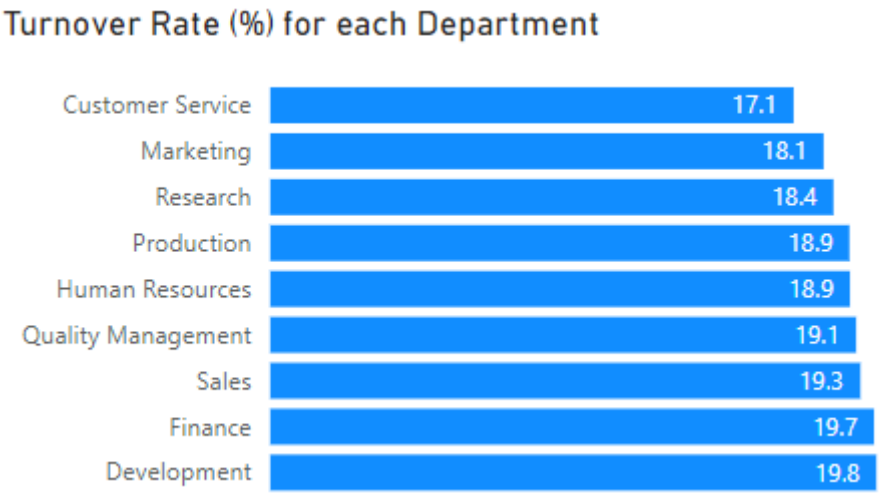

Figure 12. Turnover Rate by Department

We can see that all departments have a similar turnover rate, ranging between 17-20%. Customer Service has the lowest turnover rate at 17.1% whilst Finance and Development have the largest turnover rates at 19.7% and 19.8% respectively.

10. Employee Satisfaction Analysis

This question asks to use salary progression, however salary progression cannot be determined as stated in Question 3 of this section. By extension, this question cannot be answered.

Dashboard

A screenshot of the final dashboard is displayed on the following page. Please refer to the attached Power BI file to see the interactive dashboard. Note that due to the full data not being loaded into Power BI due to computational limitations (rather outputted tables from SQL were added manually), there is not a full degree of interactivity for every chart.

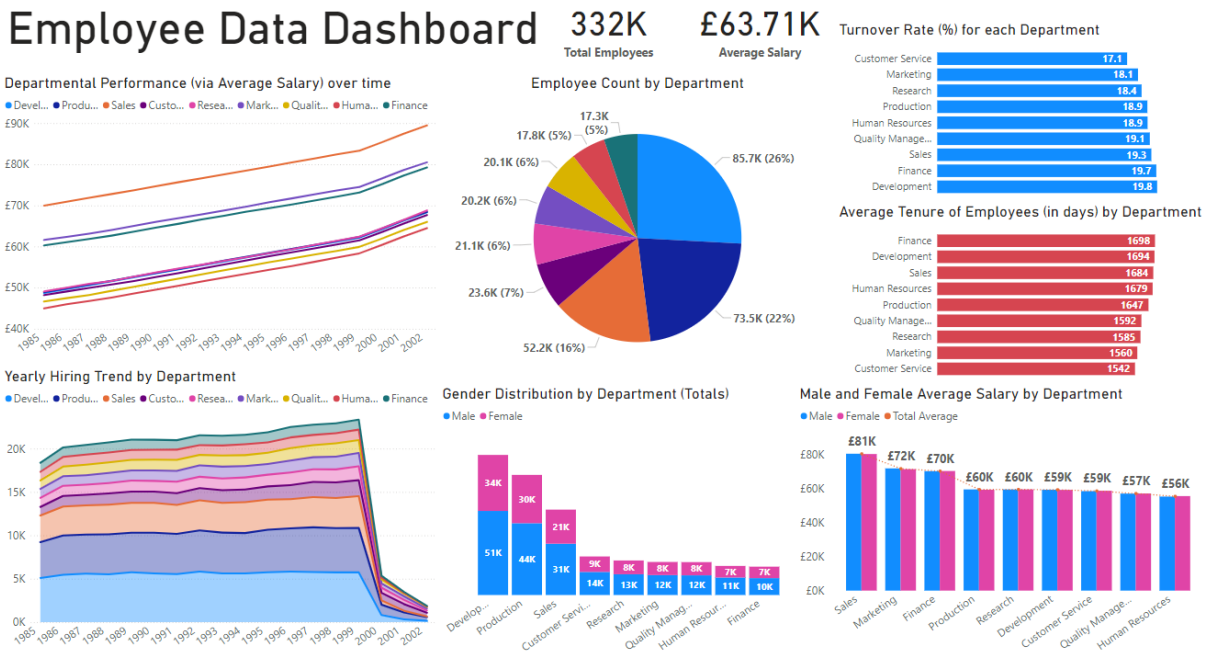

Figure 13. Screenshot of Dashboard.