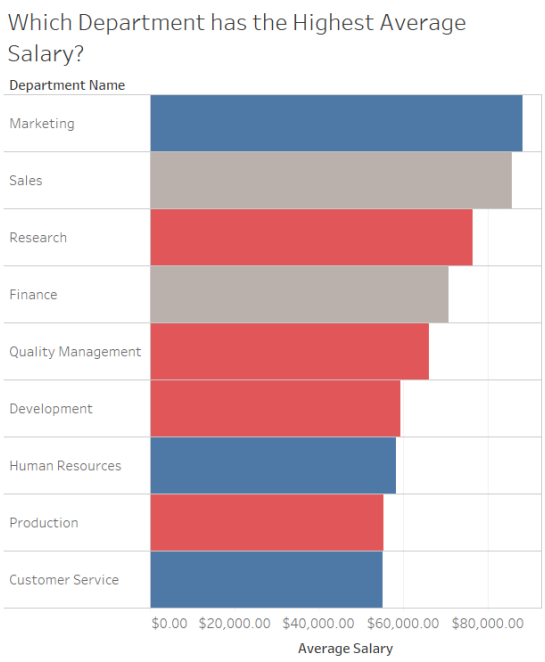

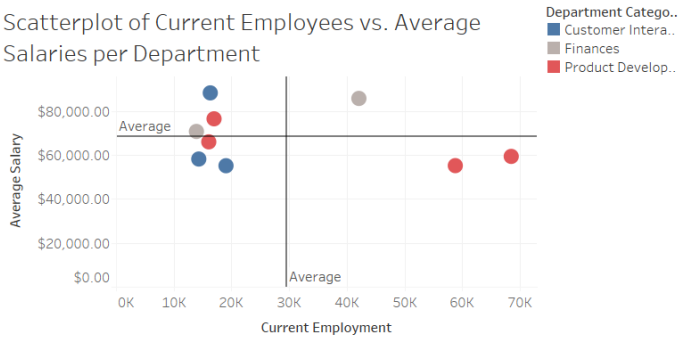

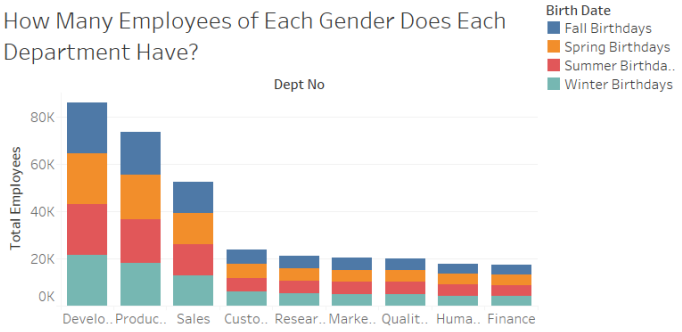

Bar Chart:

1. The story this visualization communicates is how much the average salaries between the departments can vary, and how some are closer than others so just looking at a total average in a table could be harder to tell the differences.
2. It reflects good design principles by having an easy to read chart being a bar chart and I tried to use as few colors as possible that also wouldn't be confused by people with colorblindness. I categorized the departments by the key used for the table in the top right.

Scatterplot:

1. This visualization helps show the comparison between the number of employees in each department along with their average salaries, so it is similar to the bar chart except it gives more detail. It also has an average line so you can see any variances or outliers much easier than looking at a table.
2. I used similar colors as the last bar chart and tried to keep the labels clear so that the user can tell what each point means and what the table is supposed to represent.

Stacked Bar Chart:

1. The story this shows is how each department's employees' birthdays were distributed. Although the data might not show too much, it still would be easier to look at than a table since the stacked bar chart can be useful to show parts of a whole.

2. I tried to use good design principles by grouping the colors of the stacked bar chart to limit the number of colors used. I also sorted by number of employees so that the presumably larger sections would be shown together and the smaller ones would be shown together.

Queries

1. Which department manager has the highest salary on average over the course of their employment?

mysql> SELECT departments.dept_no as DepartmentNumber, departments.dept_name as DepartmentName, dept_manager.emp_no as EmployeeNumber, first_name as FirstName, last_name as LastName, round(avg(salary),2) as AverageSalary

-> FROM departments, dept_manager, employees, salaries

-> WHERE departments.dept_no = dept_manager.dept_no and dept_manager.emp_no = employees.emp_no and dept_manager.emp_no = salaries.emp_no

-> GROUP BY dept_manager.emp_no

-> ORDER BY round(avg(salary),2) desc;

DepartmentNumber	DepartmentName	EmployeeNumber	FirstName	LastName	AverageSalary
d001	Marketing	110022	Margareta	Markovitch	89128.28
d001	Marketing	110039	Vishwani	Minakawa	87570.59
d007	Sales	111133	Hauke	Zhang	87422.13
d008	Research	111400	Arie	Staelin	87217.28
d007	Sales	111035	Przemyslawa	Kaelbling	84242.44
d006	Quality Management	110725	Peternela	Onuegbe	81301.22
d002	Finance	110085	Ebru	Alpin	72822.78
d002	Finance	110114	Isamu	Legleitner	68809.00
d006	Quality Management	110800	Sanjoy	Quadeer	67995.50
d008	Research	111534	Hilary	Kambil	65916.67
d004	Production	110344	Rosine	Cools	63609.71
d003	Human Resources	110183	Shirish	Ossenbruggen	62990.22
d009	Customer Service	111877	Xiaobin	Spinelli	62860.36
d005	Development	110511	DeForest	Hagimont	62568.17
d006	Quality Management	110854	Dung	Pesch	59734.29
d009	Customer Service	111692	Tonny	Butterworth	58223.61
d004	Production	110303	Krassimir	Wegerle	57052.33
d005	Development	110567	Leon	DasSarma	56384.31
d006	Quality Management	110765	Rutger	Hofmeyr	55320.57
d004	Production	110386	Shem	Kieras	53594.00
d003	Human Resources	110228	Karsten	Sigstam	53581.89
d009	Customer Service	111784	Marjo	Giarratana	50033.13
d009	Customer Service	111939	Yuchang	Weedman	49833.93
d004	Production	110420	Oscar	Ghazalie	46852.82

24 rows in set (0.00 sec)

2. How many employees does each department currently have?

mysql> SELECT departments.dept_no as DepartmentNumber, dept_name as DepartmentName, count(*) as CurrentEmployees

-> FROM departments, dept_emp

-> WHERE departments.dept_no = dept_emp.dept_no and emp_no in (SELECT emp_no FROM dept_emp WHERE to_date > '9000-01-01')

-> GROUP BY dept_emp.dept_no

-> ORDER BY count(*) desc;

```
2. How many employees does each department currently have?
mysql> SELECT departments.dept_no as DepartmentNumber, dept_name as DepartmentName, count(*) as CurrentEmployees
    -> FROM departments, dept_emp
    -> WHERE departments.dept_no = dept_emp.dept_no and emp_no in (SELECT emp_no FROM dept_emp WHERE to_date > '9000-01-01')
    -> GROUP BY dept_emp.dept_no
    -> ORDER BY count(*) desc;
+------------------+--------------------+------------------+
| DepartmentNumber | DepartmentName     | CurrentEmployees |
+------------------+--------------------+------------------+
| d005             | Development        |            68392 |
| d004             | Production         |            58706 |
| d007             | Sales              |            42000 |
| d009             | Customer Service   |            18978 |
| d008             | Research           |            16904 |
| d001             | Marketing          |            16252 |
| d006             | Quality Management |            15990 |
| d003             | Human Resources    |            14260 |
| d002             | Finance            |            13850 |
+------------------+--------------------+------------------+
9 rows in set (0.53 sec)
```

3. What is the average career salary of men and women at the company, if they were fired before 2000?

mysql> SELECT gender as Gender, count(*) as EmployeeCount, round(avg(salary),2) as AverageCareerSalary

-> FROM employees, salaries, dept_emp

-> WHERE employees.emp_no = salaries.emp_no and employees.emp_no = dept_emp.emp_no and employees.emp_no IN (SELECT emp_no FROM dept_emp WHERE to_date < '2000-01-01')

-> GROUP BY gender;

```
3. What is the average career salary of men and women at the company, if they were fired before 2000?
mysql> SELECT gender as Gender, count(*) as EmployeeCount, round(avg(salary),2) as AverageCareerSalary
    -> FROM employees, salaries, dept_emp
    -> WHERE employees.emp_no = salaries.emp_no and employees.emp_no = dept_emp.emp_no and employees.emp_no IN (SELECT emp_no FROM dept_emp WHERE to_date < '2000-01-01')
    -> GROUP BY gender;
+--------+---------------+---------------------+
| Gender | EmployeeCount | AverageCareerSalary |
+--------+---------------+---------------------+
| F      |        250280 |            62670.08 |
| M      |        374852 |            62877.41 |
+--------+---------------+---------------------+
2 rows in set (0.91 sec)
```