Innovations

I’ve included a new menu option (8) which gives the user three further choices.

1. The user is asked to enter the Department ID and the age of each employee in that department is returned. I used the *datediff()* function to get the difference in days between now and the employee‘s date of birth. I divided that by 365.25 and rounded the result down to get the age. The SQL query is as follows:

```
“”” Select name, ROUND(AVG(datediff(now(), dob) / 365.25)) AS avgAge
  FROM employee
  WHERE did = %s “””
```

Ref: https://www.codeproject.com/Questions/400138/How-to-calculate-age-from-date-of-birth-in-sql

2. The user is asked to enter the Department ID – the maximum salary of current employees in that department is returned. This one is straight forward enough.

```
“”” SELECT FORMAT(MAX(s.salary),2) as MAX
 FROM salary s
 JOIN employee e ON s.eid = e.eid
 WHERE e.did = %s “””
```

3. A table with average employee age, and the average salary of current employees by department is returned. This took a bit of thought! I wanted department name as opposed to ID so I joined to the dept table as well. I reused the average age query from above. And then to get the average current salary, I assumed that the maximum salary for each employee was the current salary. I grouped together all of the max salaries in a subquery, and then got the average. I ordered the table then by salary descending.

```
“”” SELECT d.did, d.name AS dept_name,
ROUND(AVG(datediff(now(), dob) / 365.25)) as avg_age,
 FORMAT(AVG(maxSal),2) AS AVG_MAX FROM (
 SELECT s.eid, MAX(s.salary) AS maxSal FROM salary s
 GROUP BY s.eid) as s
 RIGHT JOIN employee e ON s.eid = e.eid
 JOIN dept d ON e.did = d.did
 GROUP BY d.did
 ORDER BY AVG_MAX DESC;”””
```

Apart from this I’ve tried to improve the overall UX of the app, making it easier to navigate with clear precise instructions, with some attention paid to consistent formatting.