Fontys
Hogescholen

MODULE : **EDB4**
DAY : Monday
DATE : 21 Jan 2019
TIME : 10:30 – 12:00

Module	EDB4
Lecturer	THAQ / AVET / GPT / LPPR
Supervisor	THAQ / AVET / GPT / LPPR / SHLN / KHST
Room	2.70 / 2.72 / 2.76 / 2.84 / 2.86 / 2.88

Distribution of points: See exercises.
Number of pages: 6
Additional aids: All (non-human)

Distribution of points:

Ex. 1a	Ex. 1b	Ex. 2	Ex. 3	Ex. 4a	Ex. 4b
25	10	20	25	15	5

Remarks:

- **All exercises refer to the HR database as described in appendix 1.**
- You may use a 'print' or 'p' procedure instead of 'dbms_output.put_line'.
- Copy your solution into a (one) text file.
- Name the file: "**yourstudentnumber_yourname_EDB4.sql**".
- Do not forget to state your name, student number and class **inside** the text file!

Please note:

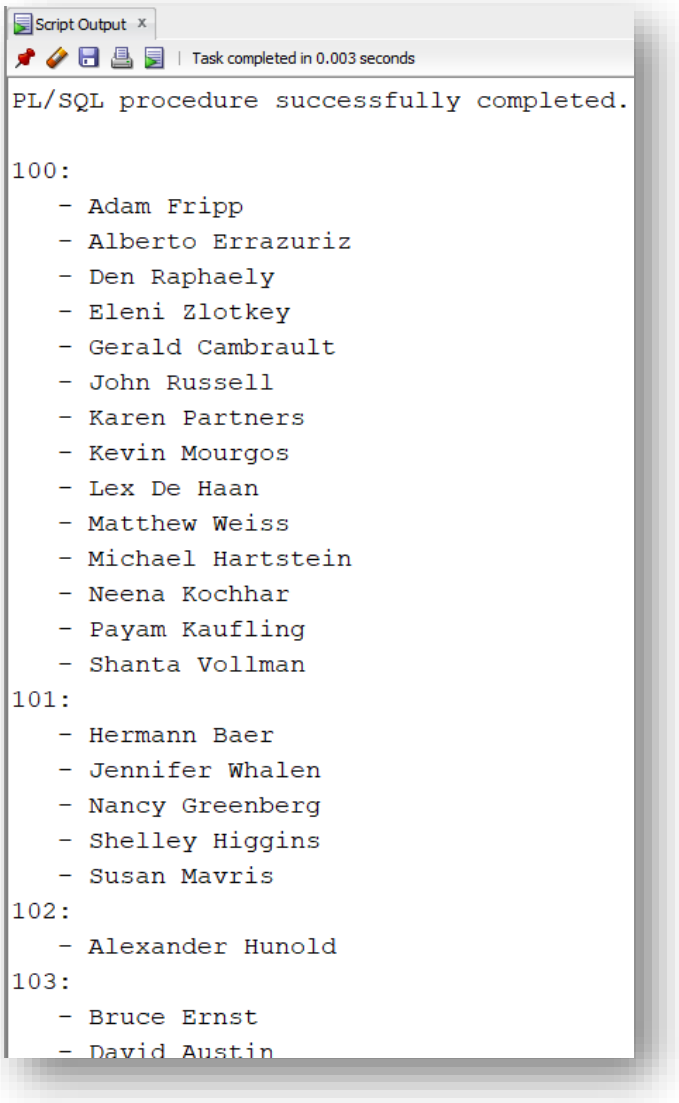
- All code must be properly indented. 1 point will be withdrawn for each solution that is not indented!
- Code will also be checked on unnecessary execution and performance. 1 point will be withdrawn for each solution that is not efficient!
- **All code must compile. If the solution does not compile, no points will be given for the particular (sub-) question!**

Success!

Exercise 1a [25 points]

Write an anonymous PL/SQL block which writes per manager (manager id) the employees (first name and last name) managed by that manager. Both the manager ids and the employee first and last names must be printed in ascending and alphabetical order respectively.

The layout of the generated list must look like the layout of the example shown below:



Exercise 1b [10 points]

Adapt the PL/SQL block of exercise 1a (make a copy first!) for the following two changes:
1. It prints not only manager id, but also manager first and last names;
2. It prints not only employee first and last names, but also the sequential number of an employee per manager.

The layout of the generated list must look like the layout of the example shown below:

Exercise 2 [20 points]

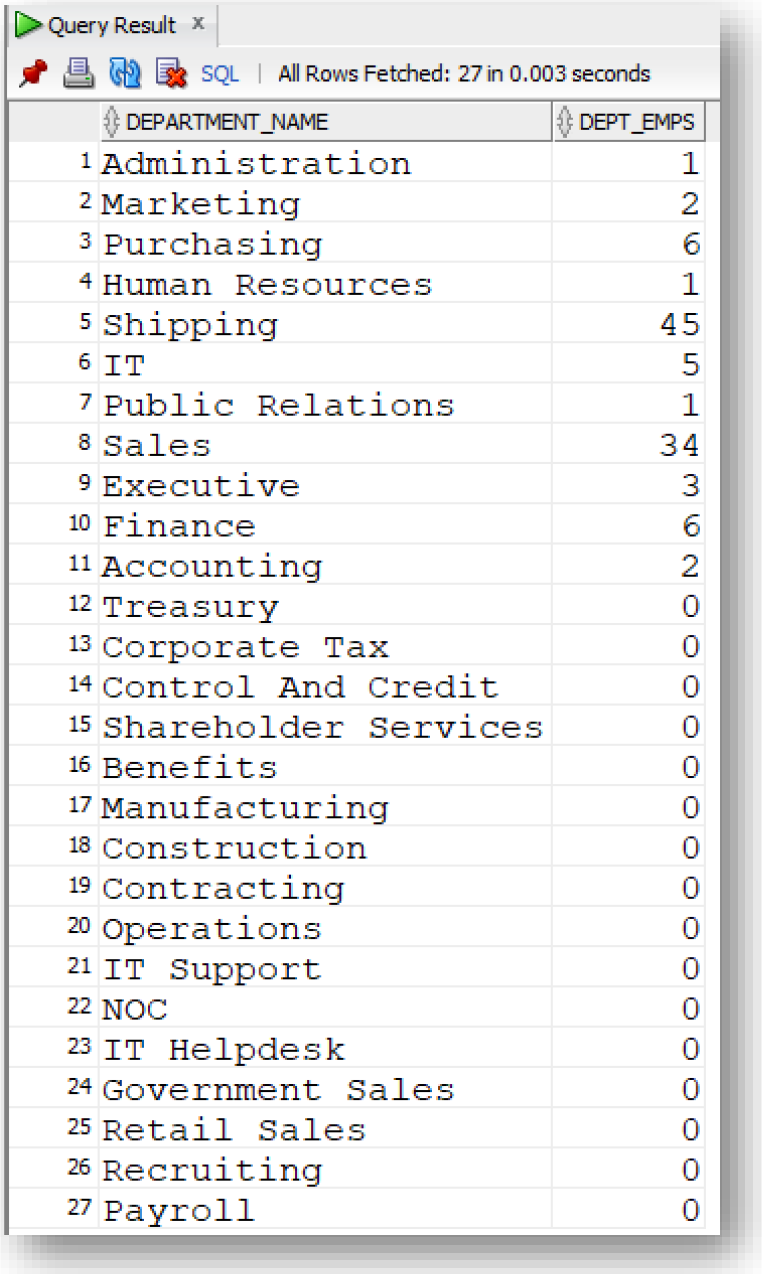
Write a stored function which can be used to simplify the query to list all the names of the departments and the number of employees working for the department.

Implement ***Dept_Emps*** stored function.

For example, the query:

```
SELECT department_name, Dept_Emps(department_id) AS dept_emps
FROM DEPARTMENTS;
```

Should result in output similar to as given below:


Query Result

SQL | All Rows Fetched: 27 in 0.003 seconds

	DEPARTMENT_NAME	DEPT_EMPS
1	Administration	1
2	Marketing	2
3	Purchasing	6
4	Human Resources	1
5	Shipping	45
6	IT	5
7	Public Relations	1
8	Sales	34
9	Executive	3
10	Finance	6
11	Accounting	2
12	Treasury	0
13	Corporate Tax	0
14	Control And Credit	0
15	Shareholder Services	0
16	Benefits	0
17	Manufacturing	0
18	Construction	0
19	Contracting	0
20	Operations	0
21	IT Support	0
22	NOC	0
23	IT Helpdesk	0
24	Government Sales	0
25	Retail Sales	0
26	Recruiting	0
27	Payroll	0

Exercise 3 [25 points]

One of the business rules which must be implemented is:
The salary of an employee may never be increased or decreased by more than 10%.
When this rule is violated an appropriate message should be displayed and the data manipulation action which caused this situation must be cancelled.

Implement this business rule.

Exercise 4a [15 points]

Create a stored procedure ***Emps_Per_City*** which returns the number of employees working in departments located in the specified city.
Use 2 parameters:

1. First parameter **p_city** to specify the city of the departments
2. Second parameter **p_num_emps** should be used to return the number of employees working in all departments of the specified city

Exercise 4b [5 points]

Write an anonymous PL/SQL block that retrieves number of employees working in departments located in the specified city.
You must use the procedure of exercise 4a.

END

Appendix 1 – Database description

All exercises refer to this HR database: