Contents

MySQL Questions

Write only the exact MySQL command for each question into the appropriate file.

Question A (MySQLQA.txt)

Show the *dept_name* of departments in which employees who were born in September work, or have worked in.

The results should be sorted alphabetically by *dept_name*.

```
+--------------------+
| dept_name          |
+--------------------+
| Development        |
| Finance            |
| Production         |
| Quality Management |
| Research           |
| Sales              |
+--------------------+
6 rows in set (0.00 sec)
```

Figure 1 Example of output required for this question

Question B (MySQLQB.txt)

Show the *title* (or titles) of employee(s) with the highest salary/salaries.

The results should be sorted alphabetically by *title*.

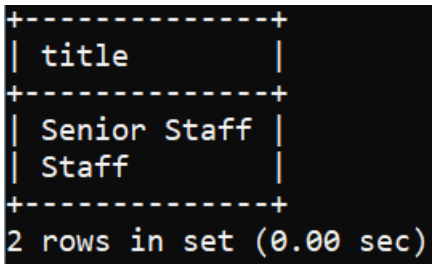


Figure 2 Example of output required for this question.

Question C (MySQLQC.txt)

For each employee show his/her *emp_no*, current *salary*, and his/her current *title*.

The results should be sorted by ascending *emp_no*.

emp_no	salary	title
10001	88958	Senior Engineer
10002	72527	Staff
10003	43311	Senior Engineer
10004	74057	Senior Engineer
10005	94692	Senior Engineer
10006	59755	Senior Engineer
10007	88070	Senior Staff
10008	52668	Assistant Engineer
10009	94409	Senior Engineer
10010	80324	Engineer
10011	56753	Staff
10012	54423	Senior Engineer
10013	68901	Senior Staff
10014	60598	Engineer

Figure 3 Example of output required for this question.

Question D (MySQLQD.txt)

Show the *emp_no*, *first_name*, *last_name*, and all *salaries* that are above the average salary of all employees.

The results should be sorted by ascending *emp_no*, and within that by ascending *salary*.

emp_no	first_name	last_name	salary
10001	Georgi	Facello	66074
10001	Georgi	Facello	66596
10001	Georgi	Facello	66961
10001	Georgi	Facello	71046
10001	Georgi	Facello	74333
10001	Georgi	Facello	75286
10001	Georgi	Facello	75994
10001	Georgi	Facello	76884
10001	Georgi	Facello	80013
10001	Georgi	Facello	81025
10001	Georgi	Facello	81097
10001	Georgi	Facello	84917
10001	Georgi	Facello	85097
10001	Georgi	Facello	85112
10001	Georgi	Facello	88958
10002	Bezalel	Simmel	65828
10002	Bezalel	Simmel	65909
10002	Bezalel	Simmel	67534

Figure 4 Example of output required for this question.

Question E (MySQLQE.txt)

Show the *emp_no*, *first_name*, *last_name*, *salary* (formatted to 2 decimal places), and a column entitled *Months on Salary* which shows the number of months an employee spent on a particular salary. The employee's current salary should be excluded.

The results should be sorted in ascending *emp_no*, and within that by ascending *salary*.

emp_no	first_name	last_name	salary	Months on Salary
10001	Georgi	Facello	60,117.00	12
10001	Georgi	Facello	62,102.00	12
10001	Georgi	Facello	66,074.00	12
10001	Georgi	Facello	66,596.00	12
10001	Georgi	Facello	66,961.00	12
10001	Georgi	Facello	71,046.00	12
10001	Georgi	Facello	74,333.00	12
10001	Georgi	Facello	75,286.00	12
10001	Georgi	Facello	75,994.00	12
10001	Georgi	Facello	76,884.00	12
10001	Georgi	Facello	80,013.00	12
10001	Georgi	Facello	81,025.00	12
10001	Georgi	Facello	81,097.00	12
10001	Georgi	Facello	84,917.00	12
10001	Georgi	Facello	85,097.00	12
10001	Georgi	Facello	85,112.00	12
10002	Bezalel	Simmel	65,828.00	12
10002	Bezalel	Simmel	65,909.00	12
10002	Bezalel	Simmel	67,534.00	12

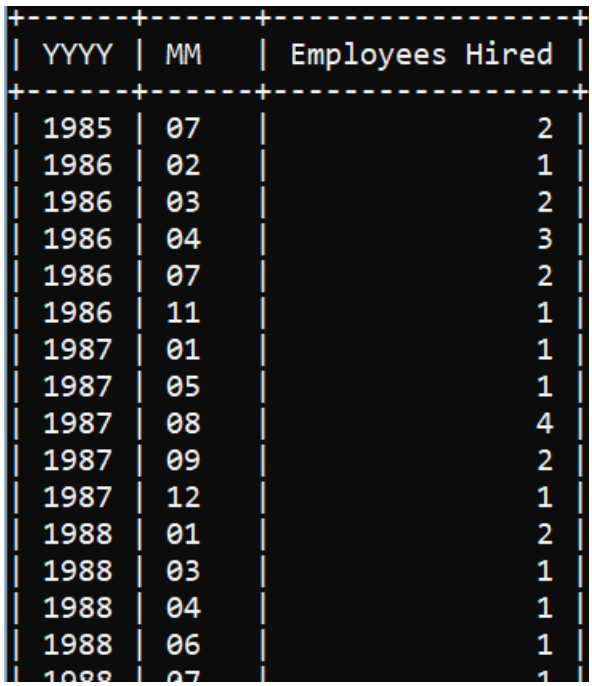
Figure 5 Example of output required for this question.

Question F (MySQLQF.txt)

Show the year as *YYYY*, the month as *MM* and a column entitled *Employees Hired* which contains the number of employees hired in that particular year and month.
However, when displaying the results 5 months should be added to each *hire_date*.
For example, a *hire_date* of September, 1985 would become February, 1986.

NOTE: The data in the table should remain unchanged.

The results should be sorted in ascending *YYYY*, within that by ascending *MM*, and within that by ascending *Employees Hired*.



YYYY	MM	Employees Hired
1985	07	2
1986	02	1
1986	03	2
1986	04	3
1986	07	2
1986	11	1
1987	01	1
1987	05	1
1987	08	4
1987	09	2
1987	12	1
1988	01	2
1988	03	1
1988	04	1
1988	06	1
1988	07	1

Figure 6 Example of output required for this question.

Neo4j Questions

Write only the exact Neo4j/Cypher command for each question into the appropriate file.

Question A (Neo4jQA.txt)

Return the average *gpa* (as *GalwayGPA*) for students in the Location *Galway*.

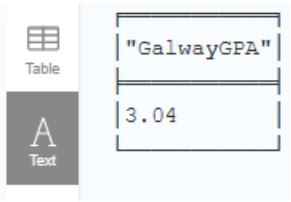


Figure 7 Example of output required for this question.

Question B (Neo4jQB.txt)

Return the average *points* (as *ATU_Avg_Points*), of all courses in the Institution *Atlantic Technological University*.

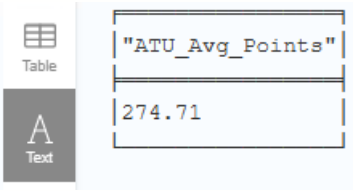


Figure 8 Example of output required for this question.

Question C (Neo4jQC.txt)

Return all friend-of-a-friends of student with the *sid* **TUS-L017** (as *FOAF*) and the name of the course the friend-of-a-friend is studying (as *FOAF_Course*).

Results should be returned in alphabetical *FOAF* order, followed by alphabetical *FOAF_Course* order.

(NOTE: A friend-of-a-friend in someone who is friends with a friend of a person, but not with the friend themselves.
In the following example, “Tadgh Robins” is a FOAF of “Fiona Murphy”, but “Clodagh Murphy” is not a FOAF of “Fiona Murphy”; because “Clodagh Murphy” has a direct friendship with “Fiona Murphy”, but “Tadgh Robins” does not).

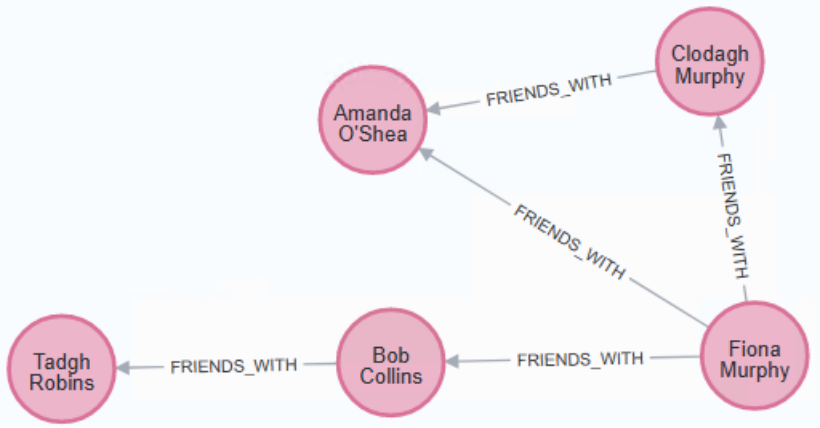


Table
Text
Code

"FOAF"	"Course"
"Brian McBrien"	"B.A. in Hotel Management"
"Carol Connolly"	"B.A. in Social Studies"
"Laura Collins"	"B.A. in Social Studies"
"Tadgh Robins"	"B.Eng. (Hons) in Industrial Engineering"

Figure 9 Example of output required for this question.

Question D (Neo4jQD.txt)

Return the *location* (as *Location*) and the number of *courses* with students studying them (as *Courses_with_Students*) for each location in the Institution "Atlantic Technological University"

Results should be returned in *Courses_with_Students* order and within that by alphabetical *Location*.

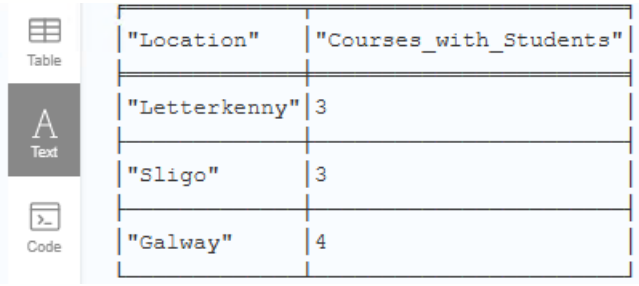


Figure 10 Example of output required for this question.

Question E (Neo4jQE.txt)

Return the course *name* (as *Name*), the course *cao* (as *CAO*), and the total *gpa* for students studying that course (as *Total_GPA*), for all courses with students studying them in the Location *Galway*.

Results should be returned by alphabetical *CAO*.

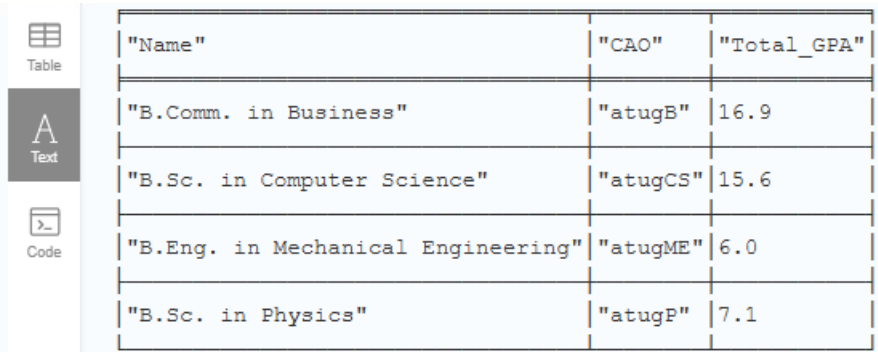


Figure 11 Example of output required for this question.

Question F (Neo4jQF.txt)

For the Institution *Atlantic Technological University*, return the *location* (as *Location*), the Course *name* (as *Course*), and a column entitled *Class_Size* that has one of the following values:

- *Tiny* if the number of students studying a course is less than 4
- *Small* if the number of students studying a course is equal to 4
- *Big* if the number of students studying a course is equal to 5
- *Large* if the number of students studying a course is greater than 6.

Results should be sorted alphabetically by *Location*, and within that alphabetically by *Course*.

Table
Text
Code

"Location"	"Course"	"Class_Size"
"Galway"	"B.Comm. in Business"	"Large"
"Galway"	"B.Eng. in Mechanical Engineering"	"Tiny"
"Galway"	"B.Sc. in Computer Science"	"Big"
"Galway"	"B.Sc. in Physics"	"Tiny"
"Letterkenny"	"B.A (Hons) in Accounting"	"Tiny"
"Letterkenny"	"B.A. in Hotel Management"	"Tiny"
"Letterkenny"	"B.Sc. (Hons) in Physics"	"Tiny"
"Sligo"	"B.A (Hons) in Accounting"	"Small"
"Sligo"	"B.Eng (Hons) in Industrial Engineering"	"Tiny"
"Sligo"	"B.Sc. (Hons) in Physics"	"Tiny"

Figure 12 Example of output required for this question.