Exercises: Data Definition and Data Types

This document defines the **exercise assignments** for the "Databases Basics - MySQL" course @ Software University.

Mr. Bodrog is a greedy small goblin who is in charge of Gringotts – the biggest wizard bank. His most precious possession is a small database of the deposits in the wizard's world. Taking money is his hobby. He wants your money as well but unfortunately you are not a wizard. The only magic you know is how to work with databases. That's how you got access to the precious data. Mr. Bodrog wants you to send him some reports otherwise he will send a pack of hungry werewolves after you. You don't want to confront pack of hungry werewolves, do you?

Problem 1. Records' Count

Import the database and send the total count of records to Mr. Bodrog. Make sure nothing got lost.

Example:

count
162

Problem 2. Longest Magic Wand

Select the size of the longest magic wand. Rename the new column appropriately.

Example:

longest_magic_wand
31

Problem 3. Longest Magic Wand per Deposit Groups

For wizards in each deposit group show the longest magic wand. Rename the new column appropriately.

Example:

deposit_group	longest_magic_wand
Human Pride	30
…	…

Problem 4. * Smallest Deposit Group per Magic Wand Size

Select the deposit group with the lowest average wand size.

Example:

deposit_group
Troll Chest
Venomous Tongue

Problem 5. Deposits Sum

Select all deposit groups and its total deposit sum.

Example:

deposit_group	total_sum
Human Pride	1041291.52
…	…

Problem 6. Deposits Sum for Ollivander family

Select all deposit groups and its total deposit sum but only for the wizards who has their magic wand crafted by Ollivander family.

Example:

deposit_group	total_sum
Human Pride	188366.86
…	…

Problem 7. Deposits Filter

Select all deposit groups and its total deposit sum but only for the wizards who has their magic wand crafted by Ollivander family. After this filter total deposit amounts lower than 150000. Order by total deposit amount in descending order.

Example:

deposit_group	total_sum
Troll Chest	126585.18
…	…

Problem 8. Deposit charge

Create a query that selects:

- **Deposit group**
- **Magic wand creator**
- Minimum **deposit charge** for each group

Select the data in ascending order by **MagicWandCreator** and **DepositGroup**.

Example:

deposit_group	magic_wand_creator	min_deposit_charge
Blue Phoenix	Antioch Peverell	30.00
…	…	

Problem 9. Age Groups

Write down a query that creates 7 different groups based on their **age**.

Age groups should be as follows:

- [0-10]
- [11-20]
- [21-30]
- [31-40]
- [41-50]
- [51-60]
- [61+]

The query should return

- **Age groups**
- **Count** of wizards in it

Example:

age_group	wizard_count
[11-20]	21
…	…

Problem 10. First Letter

Write a query that returns all unique wizard first letters of their **first names** only if they have deposit of type Troll Chest. Order them alphabetically. Use GROUP BY for uniqueness.

Example:

first_letter
A
…

Problem 11. Average Interest

Mr. Bodrog is highly interested in profitability. He wants to know the average interest of all **deposits groups** split by whether the deposit has **expired or not**. But that's not all. He wants you to select deposits with **start date** after 01/01/1985. Order the data descending by Deposit Group and ascending by Expiration Flag.

The output should consist of the following columns:

Example:

deposit_group	is_deposit_expired	average_interest
Venomous Tongue	0	16.698947
…	…	

Problem 12. * Rich Wizard, Poor Wizard

Mr. Bodrog definitely likes his werewolves more than you. This is your last chance to survive! Give him some data to play his favorite game Rich Wizard, Poor Wizard. The rules are simple: You compare the deposits of every wizard with the wizard after him. If a wizard is the last one in the database, simply **ignore it**. At the end you have to sum the difference between the deposits.

Host Wizard	Host Wizard Deposit	Guest Wizard	Guest Wizard Deposit	Difference
Harry	10 000	Tom	12 000	-2000
Tom	12 000	…	…	…

At the end your query should return a single value: the SUM of all differences.

Example:

sum_difference
44393.97

Problem 13. Employees Minimum Salaries

That's it! You no longer work for Mr. Bodrog. You have decided to find a proper job as an analyst in SoftUni. It's not a surprise that you will use the **soft_uni** database. Things get more exciting here!

Select the minimum salary for departments with ID (2,5,7) but only for those who are hire after 01/01/2000. Your query should return:

- **department_id**
- **average_salary**

Example:

department_id	minimum_salary
2	25000.00
…	…

Problem 14. Employees Average Salaries

Select all employees who earn more than 30000 **into a new table.** Then delete all employees who has **manager_id = 42**; Then increase the salaries of all employees with **department_id =1** with 5000. Finally, select the average salaries in each department.

Example:

department_id	manager_id
1	45166.6666
…	…

Problem 15. Employees Maximum Salaries

Find the max salary for each department. Filter those which have max salaries not in the range 30000 and 70000;

Example:

department_id	max_salary
2	29800.00
…	…

Problem 16. Employees Count Salaries

Count the salaries of all employees who don't have a manager.

Example:

count
4
…

Problem 17. *3rd Highest Salary

Find the third highest salary in each department if there is such.

Example:

department_id	third_highest_salary
2	25000.00
…	…

Problem 18. **Salary Challenge

Write a query that returns

- **first_name**
- **last_name**
- **department_id**

for all employees who have salary higher than the average salary of their respective departments. Select only the first 10 rows. Order by **department_id.**

Example:

first_name	last_name	department_id
2	25000.00	
…	…	