

SQL EXERCISES 2

READ THIS FIRST! READ THIS FIRST!

- Create a Word document with the name **DM_SQL_Exercises_2_YourFamilyName.docx**
- For all your SQL queries, please copy and paste both the **SQL statement** and the **result to** your Word document. *Do not worry about any formatting.* Finally, submit the Word document to Moodle.

1. List all the male students (student number, surname, first name, gender). Sort the result by student number in ascending order.
2. List all the female students (first name, surname, student number, gender, city) who live in Espoo. Sort the result by (surname, first name, student number) in ascending order.

NOTE: A city name might have been saved in the database *either in Finnish or Swedish*.
For example, both 'Vantaa' and 'Vanda' are names of the same city.

3. List all the students (name, city) who live in the capital city of Finland. Concatenate surname, comma, and first name as "Name". Sort the result by "Name" in ascending order.
4. List all the students (surname, first name, city) who do not live in the capital city of Finland. Sort the result by (surname, first name) in ascending order.
5. List all the students (postcode, student number). Sort the result by (postcode, student number) in ascending order.
6. List all the postcodes of the areas where students live, without any duplicate entries. Sort the result in ascending order.
7. List all the students (city, postcode, surname, first name), who either live in the capital city of Finland within the postcode area 00400 – 00600 or who live in Espoo. Sort the result by (city, postcode, surname, first name) in ascending order.
8. List all the students (postcode, city, surname, first name), who do not live within the postcode area 00600 – 03000. Sort the result by (postcode, city, surname, first name) in ascending order.
9. List all the misconduct types (description) where the description starts with 'A', 'C', 'F', or 'R'. Sort the result in by description ascending order.
10. List all the misconduct types (description) where the description does not contain any of the following words: 'data', 'research', 'test'. Sort the result by description in ascending order.
11. List all the teachers' surnames that end with 'ka' or 'ko'. Sort the result in ascending order.
12. List all the course instances (course code, instance number, start date, end date) where the end date is missing. Sort the result by (course code, instance number) in ascending order.
13. List all the course grade rows (course code, instance number, student number, grade) where the grade value is missing. Sort the result by (course code, instance number, student number) in ascending order.

14. List all the known course grades (course code, instance number, student number, grade) from the course 'a290'. Sort the result by (course code, grade, student number) in ascending order.
15. Write a query that lists those teachers whose new salary would be above 18 800 €, if they would get a rise of 500 € and an additional 15 % salary increase to the old salary.

Display surname, first name, new salary, and old salary. In the query, rename the salary columns as "New salary" and "Old salary". Sort the result by (new salary, surname, first name) in descending order.

16. List all the students (surname, first name, student number, gender). Display either 'male' or 'female' as the gender. Sort the result by (surname, first name, student number) in ascending order.

Tip: Use the CASE expression to choose what to display in the column 'gender'.

The following task is not advanced, but to figure out the search condition might take a while ...

17. List all the course instances (start date, end date, course code, instance number) that have been going on during the timespan **1.12.2007** and **15.2.2008**. That is, at least one day of the course has been between these two dates.

Sort the result by (start date, course code, instance number) in ascending order.

Hints:

- ✓ If you feel unsure, see the sample query result to figure out for more details.
- ✓ The WHERE condition should be a short one. Do not try to handle all possible situations with separate tests.
- ✓ To solve this problem, you should first try to figure out the different types of situations. For example, you can visualize different cases in the following way (the figure is not complete yet):

