# Examples of tasks on SQL:

1. Show all info about the employee with ID 8.

**SELECT \* FROM employees**

**WHERE ID = 8;**

1. Show the list of first and last names of the employees from London.

**SELECT First name, Last name, City**

**FROM employees**

**WHERE city = ‘London’;**

1. Show the list of first and last names of the employees whose first name begins with letter A.

**SELECT First name, Last name**

**FROM employees**

**WHERE First name LIKE =‘A%’;**

1. Show the list of first, last names and ages of the employees whose age is greater than 55. The result should be sorted by last name.

**SELECT First name, Last name, Age**

**FROM employees**

**WHERE Age > 55**

**ORDER BY Last name;**

1. Calculate the count of employees from London.

**SELECT COUNT (\*)**

**FROM employees**

**WHERE city = ‘London’;**

1. Calculate the greatest, the smallest and the average age among the employees from London.

**SELECT City,**

**MAX (Age),**

**MIN (Age),**

**AVG (Age)**

**FROM employees**

**WHERE city = ‘London’;**

1. Calculate the greatest, the smallest and the average age of the employees for each city.

**SELECT City,**

**MAX (Age),**

**MIN (Age),**

**AVG (Age)**

**FROM employees;**

1. Show the list of cities in which the average age of employees is greater than 60 (the average age is also to be shown)

**SELECT city FROM employees**

**HAVING AVG Age > 60;**

1. Show the first and last name(s) of the eldest employee(s). Use a subquery.

**SELECT First name, Last name**

**FROM employees**

**WHERE Birthdate = (**

**SELECT MIN (Birthdate)**

**FROM employees**

**);**

1. Show first, last names and ages of 3 eldest employees.

**SELECT TOP 3 First name, Last name, Age**

**FROM employees**

**WHERE Birthdate**

**ORDER BY Birthdate ASC;**

1. Show the list of all cities where the employees are from.

**SELECT DISTINCT city**

**FROM employees;**

1. Show first, last names and dates of birth of the employees who celebrate their birthdays this month.

**SELECT First name, Last name, Birthdate**

**FROM employees**

**WHERE DATEPART(MM, Birthdate) = 1;**

1. Show first and last names of the employees who used to serve orders shipped to Madrid.
2. Show first and last names of the employees as well as the count of orders each of them have received during the year 1997 (use left join).
3. Show first and last names of the employees as well as the count of orders each of them have received during the year 1997 (use a subquery).
4. Show first and last names of the employees as well as the count of their orders shipped after required date during the year 1997 (use left join).
5. Show the count of orders made by each customer from France.
6. Show the list of french customers’ names who have made more than one order (use grouping).
7. Show the list of french customers’ names who have made more than one order (use a subquery).
8. Show the list of customers’ names who used to order the ‘Tofu’ product (use a subquery).
9. \*Show the list of customers’ names who used to order the ‘Tofu’ product, along with the total amount of the product they have ordered and with the total sum for ordered product calculated.
10. \*Show the list of french customers’ names who used to order non-french products (use left join).
11. \*Show the list of french customers’ names who used to order non-french products (use a subquery).
12. \*Show the list of french customers’ names who used to order french products.
13. \*Show the total ordering sum calculated for each country of customer.
14. \*Show the total ordering sums calculated for each customer’s country for domestic and non-domestic products separately (e.g.: “France – French products ordered – Non-french products ordered” and so on for each country).
15. \*Show the list of product categories along with total ordering sums calculated for the orders made for the products of each category, during the year 1997.
16. \*Show the list of product names along with unit prices and the history of unit prices taken from the orders (show ‘Product name – Unit price – Historical price’). The duplicate records should be eliminated. If no orders were made for a certain product, then the result for this product should look like ‘Product name – Unit price – NULL’. Sort the list by the product name.
17. \*Show the list of employees’ names along with names of their chiefs (use left join with the same table).
18. \*Show the list of cities where employees and customers are from and where orders have been made to. Duplicates should be eliminated.
19. \*Insert 5 new records into Employees table. Fill in the following fields: LastName, FirstName, BirthDate, HireDate, Address, City, Country, Notes. The Notes field should contain your own name (to distinguish your records from the ones inserted by other trainees).
20. \*Fetch the records you have inserted by the SELECT statement
21. \*Change the City field in one of your records using the UPDATE statement (first run the SELECT statement to check whether you are updating the appropriate records!).
22. \*Change the HireDate field in all your records to current date (first run the SELECT statement to check whether you are updating the appropriate records!).
23. \*Delete one of your records (first run the SELECT statement to check whether you are deleting the appropriate record!).

# Home works

1. Write SQL statements for the examples marked with (\*) above.
2. Write you own 20 examples covering all types of queries mentioned in the lecture that are cases: where, like, order by, group by, count(\*), count(<field>), sum, having, order by, inner join, left (right) join, subquery; delete, insert (+results of select), update

**Cautions**:

1. When using INSERT commands, fill some text field with your own name.
2. When using UPDATE or DELETE commands, only your records should be affected.