# SELECT queries

A query finding all students,

SELECT \*

FROM students

A query finding all teachers,

SELECT \*

FROM teachers

A query finding only first names of teachers,

SELECT split\_part(name, ' ', 1) AS first\_name

FROM teachers;

A query finding only surnames and emails of students,

SELECT surname, email

FROM students;

# WHERE clause

Finding all students whose names start with the letter **A**,

SELECT \*

FROM students

WHERE name LIKE 'A%';

Finding teachers with a salary above PLN 1900,

SELECT \*

FROM teachers

WHERE pay > 1900;

Finding marks above 4,

SELECT \*

FROM marks

WHERE mark > 4;

Finding the teacher whose name is **Bryan Cubes** (note down the teacher's id on a piece of paper),

SELECT \*

FROM teachers

WHERE name = 'Bryan Cubes'

Id = 3

Finding marks given by Bryan (use the id from the previous point — the marks you seek will have this id in their **teacher\_id** field).

SELECT \*

FROM marks

WHERE teacher\_id = 3;

# AND and OR clauses

A query finding the data of a student whose name is Damian and surname is Forrester,

SELECT \*

FROM students

WHERE name = 'Damian' AND surname = 'Forrester';

Student\_id = 3

A query finding Damian Forrester's marks higher than 3,

SELECT \*

FROM marks

WHERE student\_id = 3 AND mark > 3;

All students with names starting from D or B.

SELECT \*

FROM students

WHERE name LIKE 'D%' OR name LIKE 'B%';

# ORDER BY clause

A query finding marks given by Clara Oakley, ordered from high to low,

SELECT \*

FROM teachers

WHERE name = 'Clara Oakley';

teacher\_id = 4

SELECT mark

FROM marks

WHERE teacher\_id = 4

ORDER BY mark DESC;

A query finding all students ordered alphabetically by surname,

SELECT \*

FROM students

ORDER BY surname ASC;

A query finding all marks of the student whose email is **bertram.adams@yahoo.com** from high to low.

SELECT \*

FROM students

WHERE email = 'bertram.adams@yahoo.com';

Student\_id = 1

SELECT mark

FROM marks

WHERE student\_id = 1

ORDER BY mark DESC;

# Adding new teacher

Try to add a new teacher to the database with the following data:

INSERT INTO teachers VALUES(2, 'John Koval', 1300); - adding not worked. Key (teacher\_id)=(2) already exists.

Add the teacher from the previous point, specifying only his name and salary. Do not give the primary key (**teacher\_id** field),

INSERT INTO teachers (name, pay)

VALUES ('John Koval', 1300);

Load all teachers. What primary key has been assigned to Jan Kowalski?

SELECT \*

FROM teachers; - Yes, primary key has been assigned.

Try adding a new teacher by giving all fields (together with the primary key – **teacher\_id** field). But this time as the **teacher\_id** give the value that does not yet exist in the table (e.g. greater by one than the last value in the filed).

INSERT INTO teachers

VALUES(7, 'Bilbo Baggins', 9999)

# Adding data

Add a new class and assign it the newly-added teacher (**main\_teacher\_id** column).

INSERT INTO classes

VALUES(6, 'IX A', 7);

Add 5 new students, assigning them to the new class (**class\_id** column). Try to do this with a single SQL query.

INSERT INTO students

VALUES

(21, 'Legolas', '???', 'legolas@forrest.com', 6),

(22, 'Gimli', '???', 'gimli@mine.com', 6),

(23, 'Gandalf', '???', 'gandalf@sorcere.com', 6),

(24, 'Aragorn', '???', 'aragorn@gondor.com', 6),

(25, 'Glum', 'Smeagol', 'glum@myprecious.com', 6);