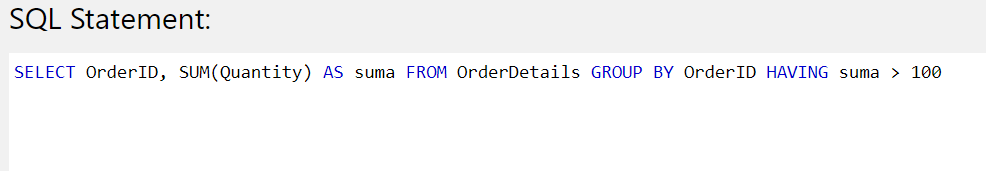
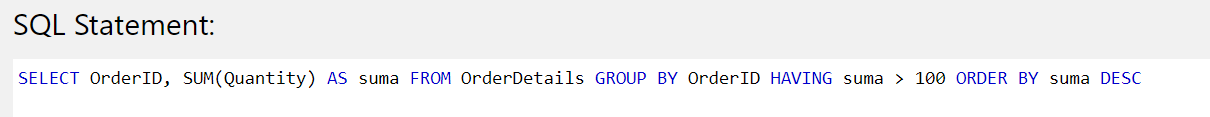
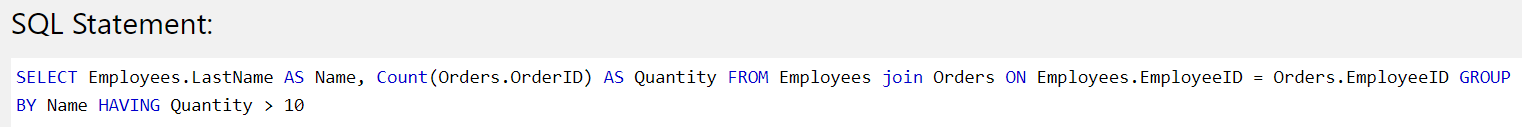
***1 задание:***

* Найти OrderID и количество всех продуктов в ордере (Quantity) из таблицы OrderDetails, где количество всех продуктов больше 100

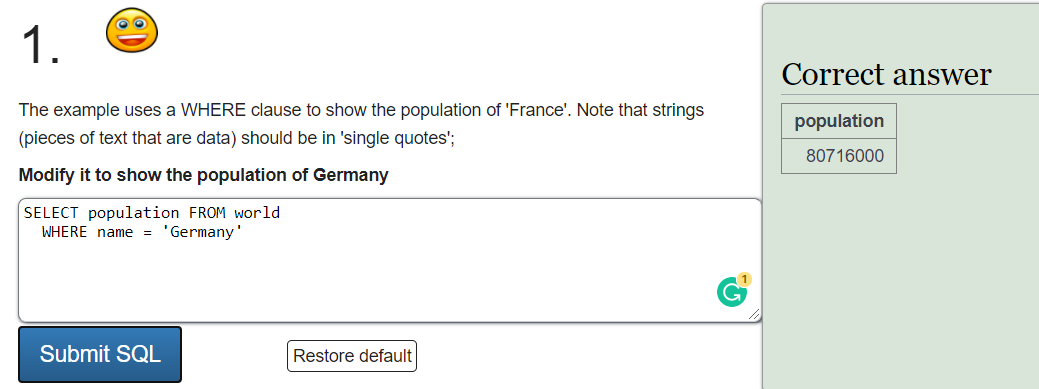


* Предыдущий пункт отсортировать по количеству от большего к меньшему
* Найти Employees.LastName employees которые имеют более 10 ордеров. Таблица Orders содержит EmployeeID, нужно использовать INNER JOIN чтобы соединить с таблицей Employees и сгруппировать по Orders.EmployeeID.

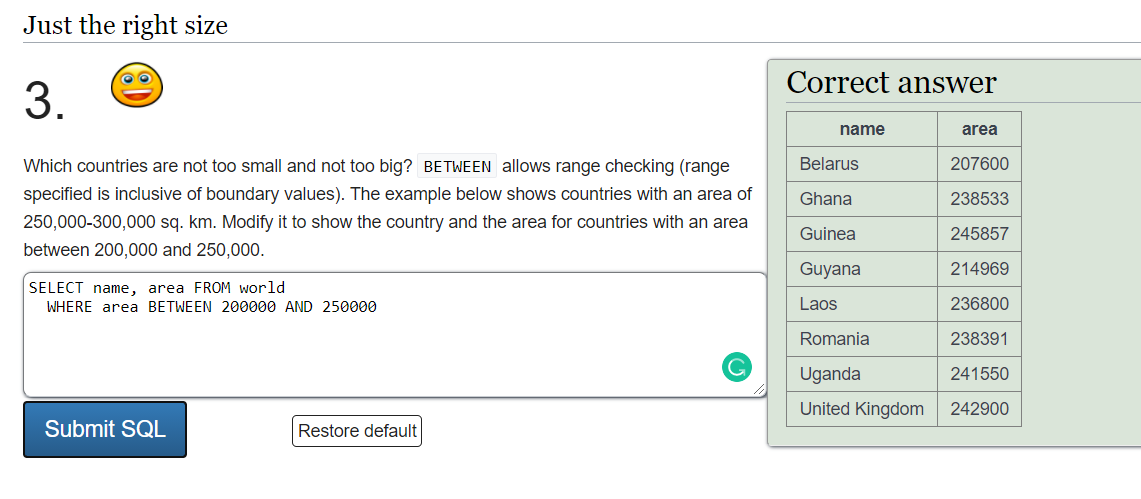
***2 задание:***

Сделать практические задания:

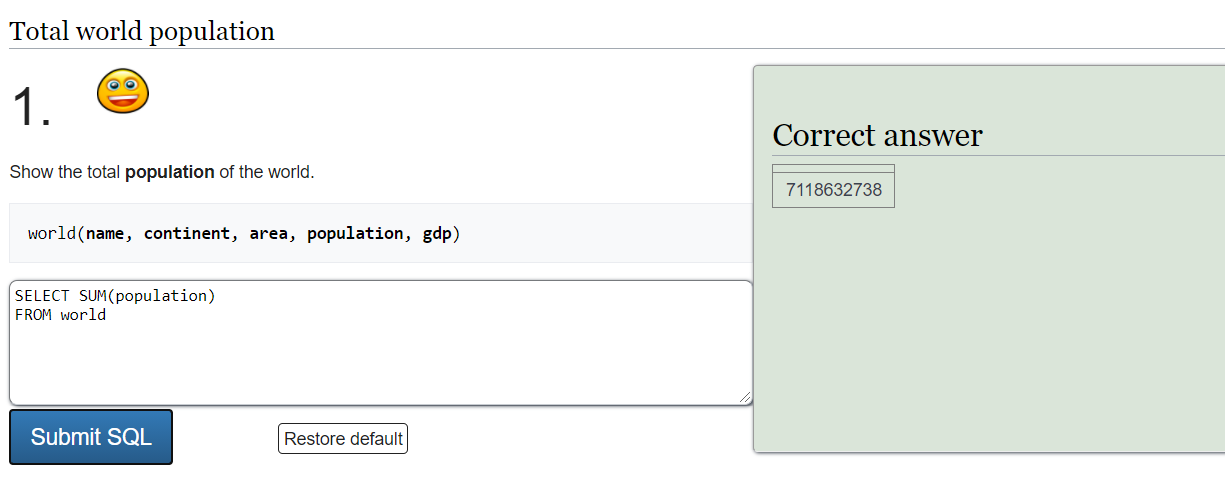
* [SELECT базовый](https://sqlzoo.net/wiki/SELECT_basics)

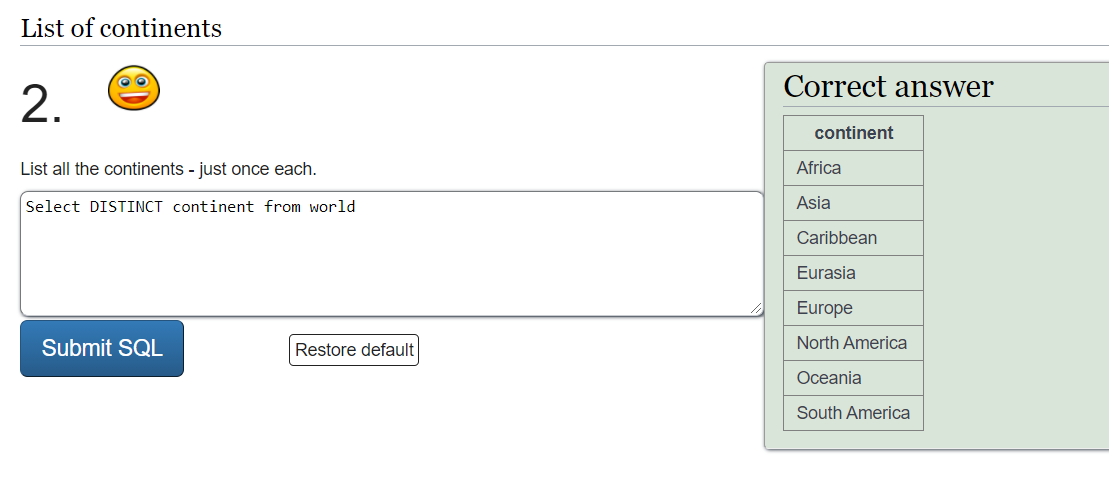




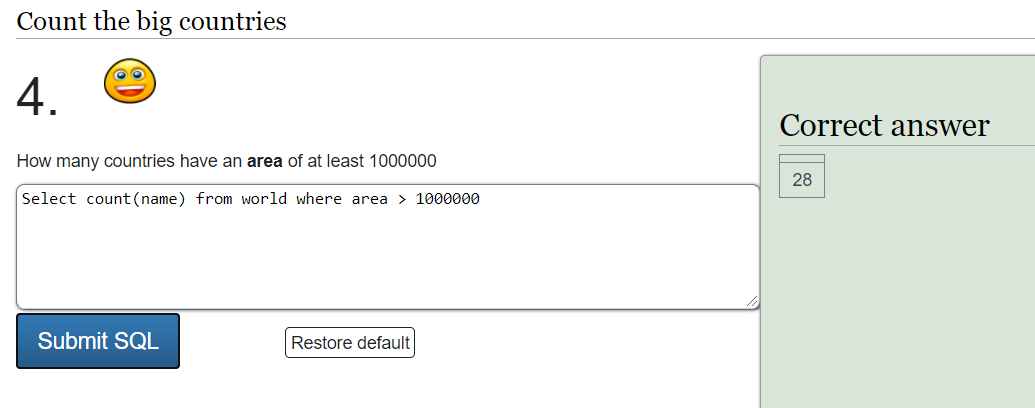


[SUM и COUNT](https://sqlzoo.net/wiki/SUM_and_COUNT)

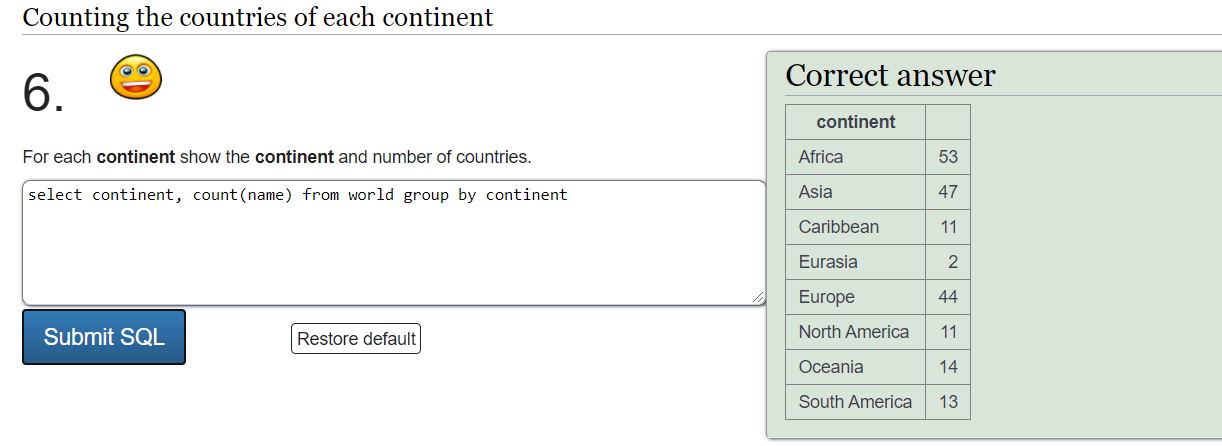


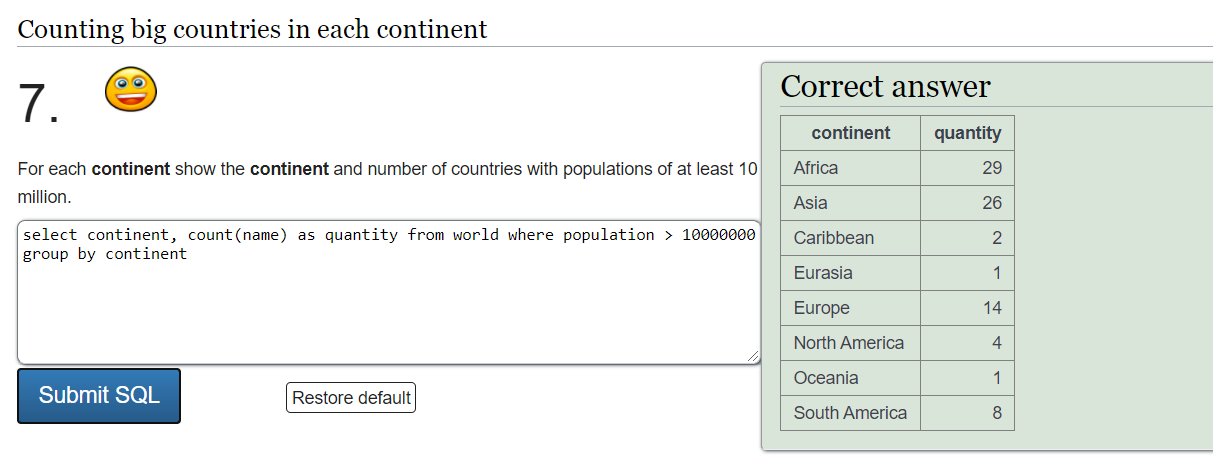


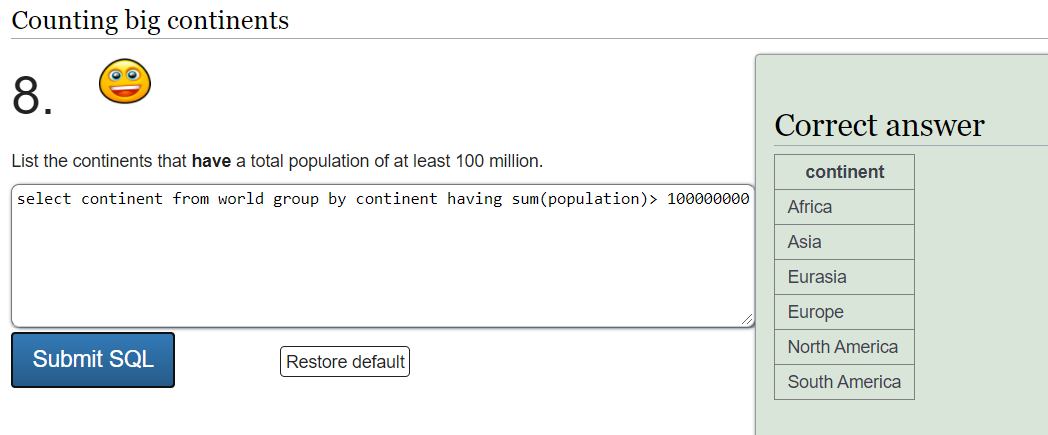






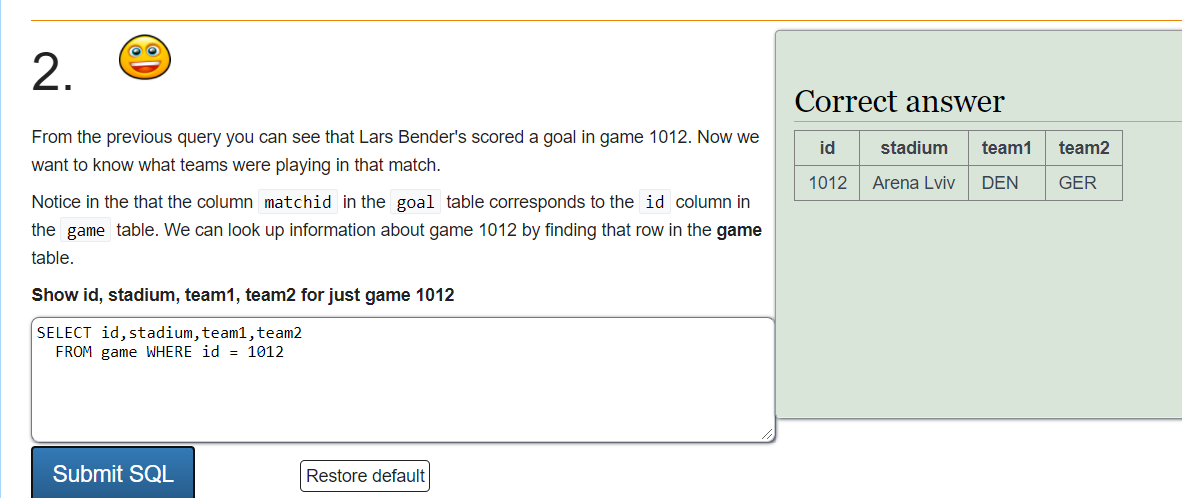


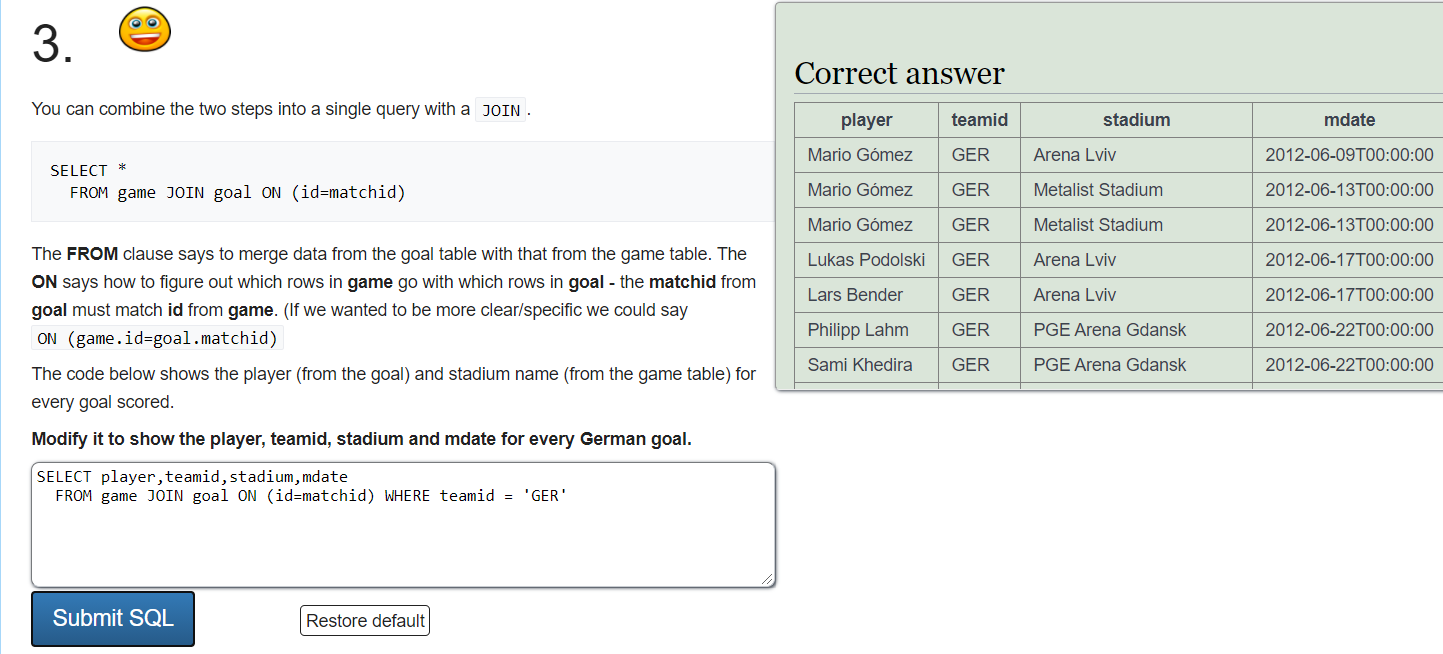


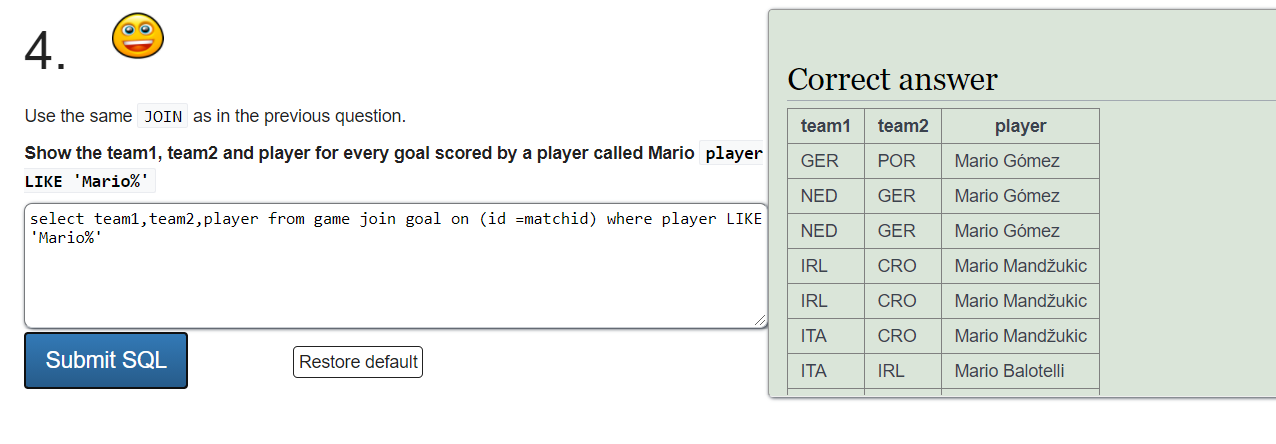


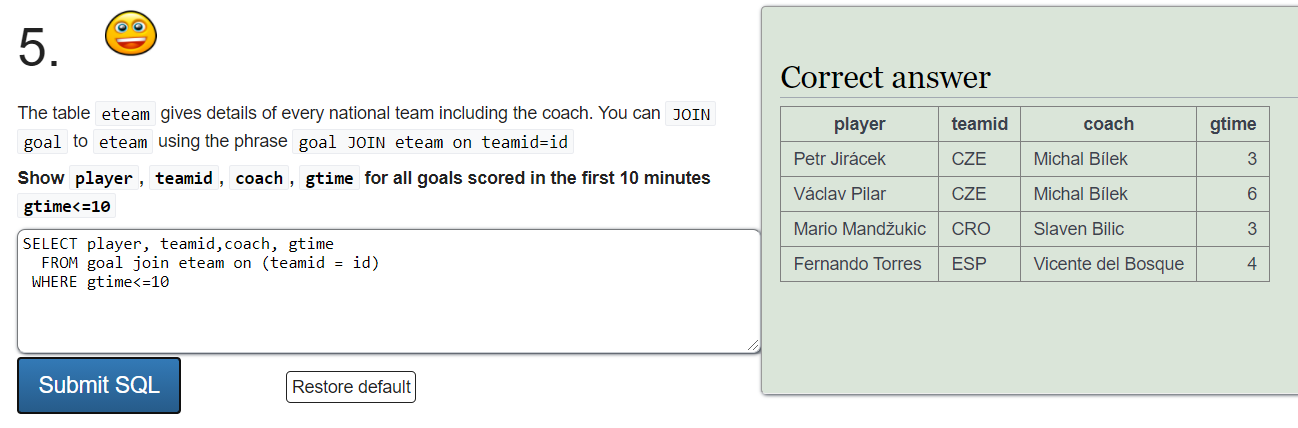
JOIN

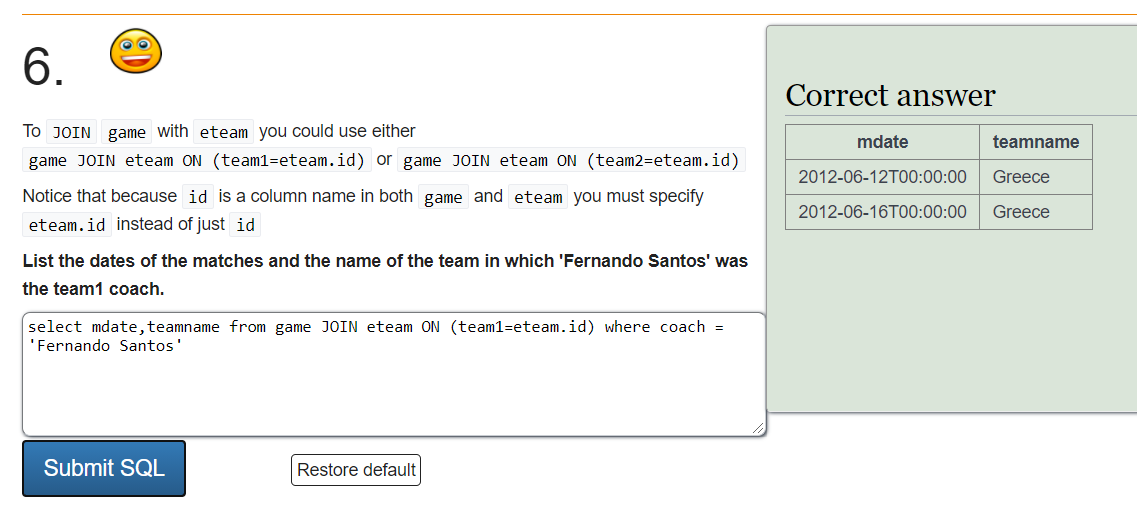


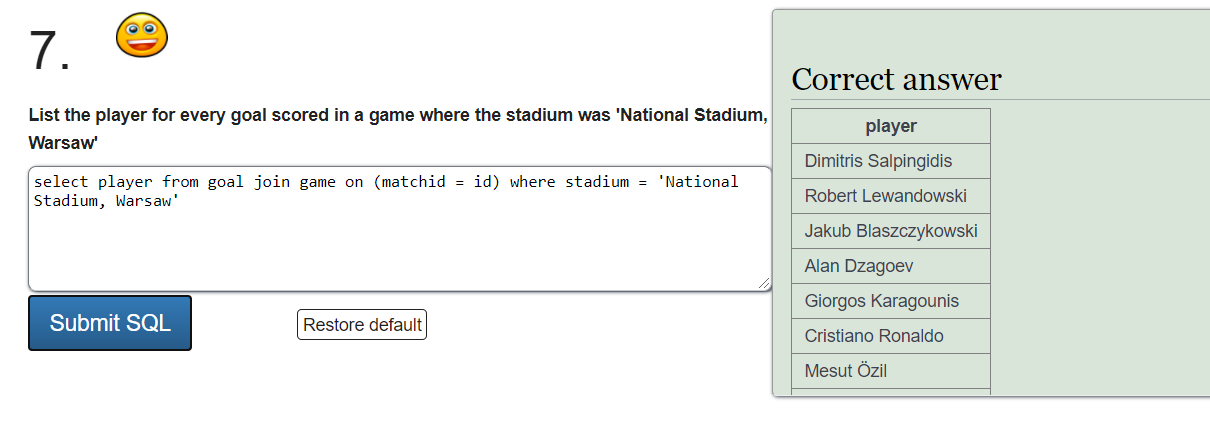


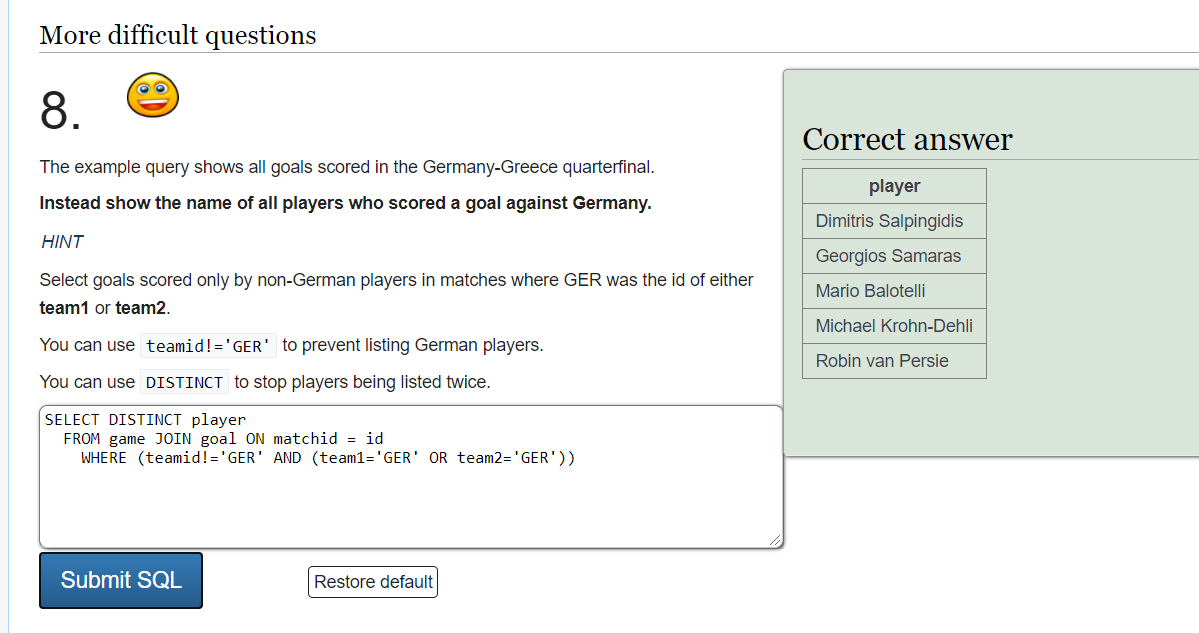


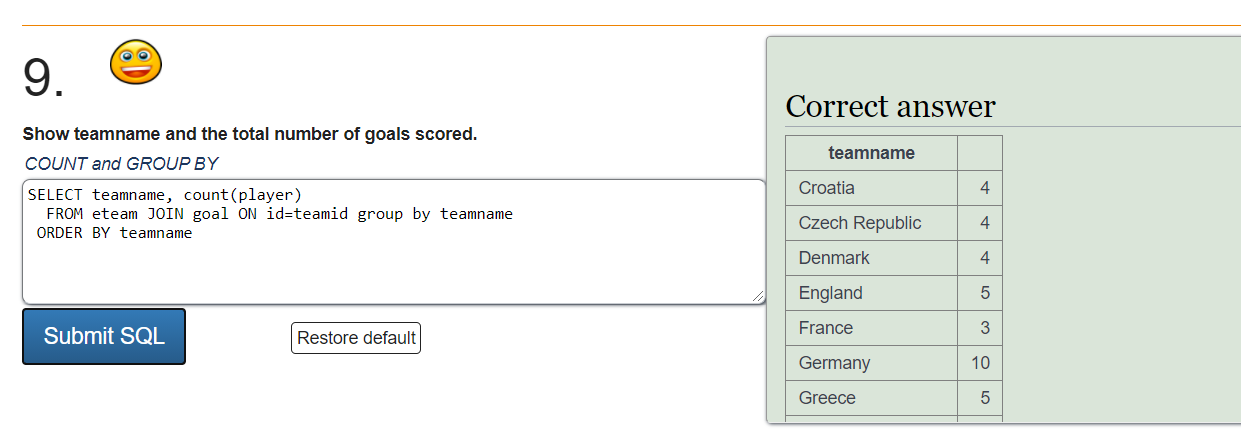


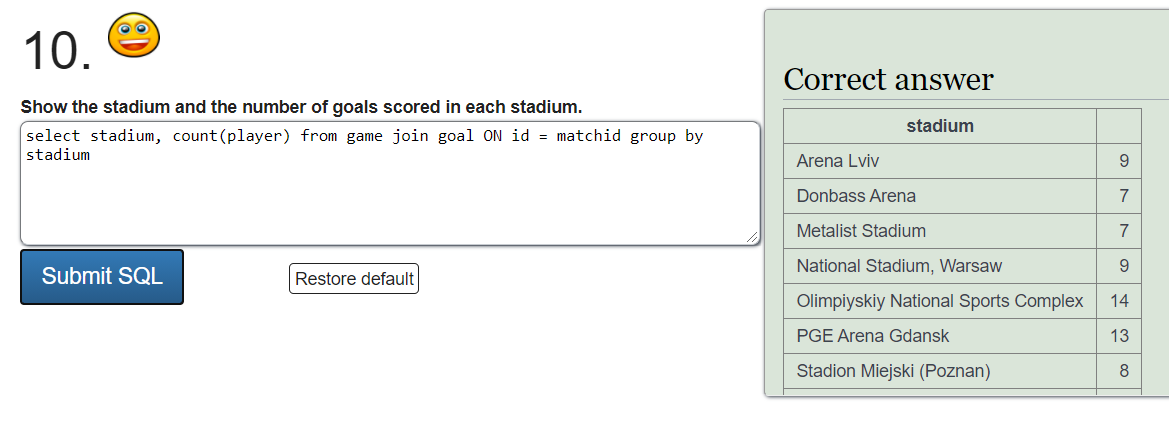


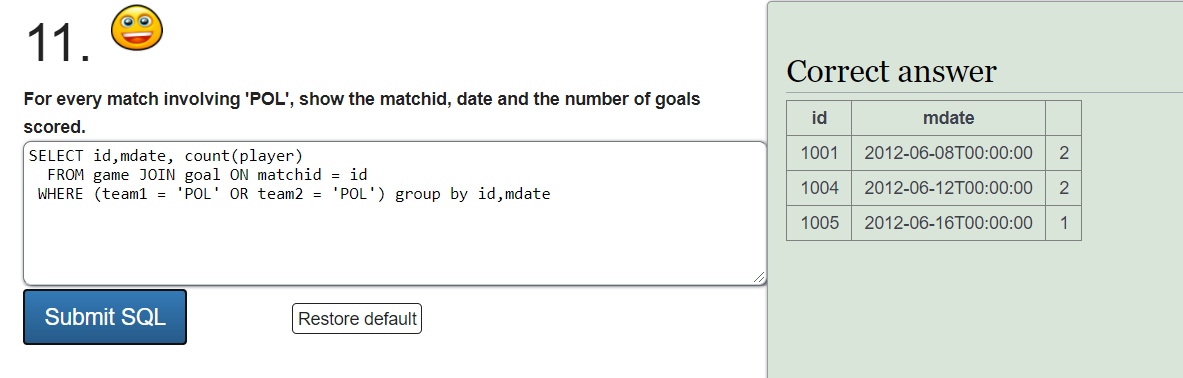


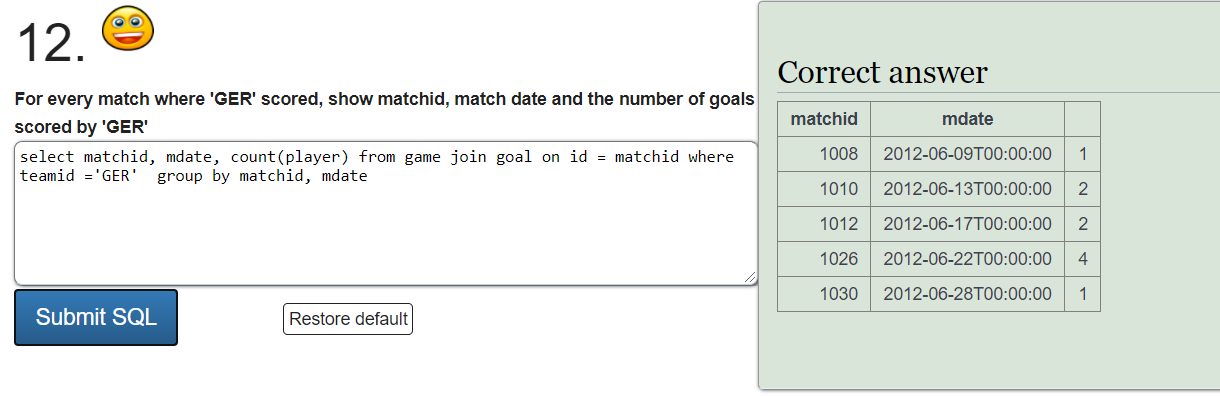


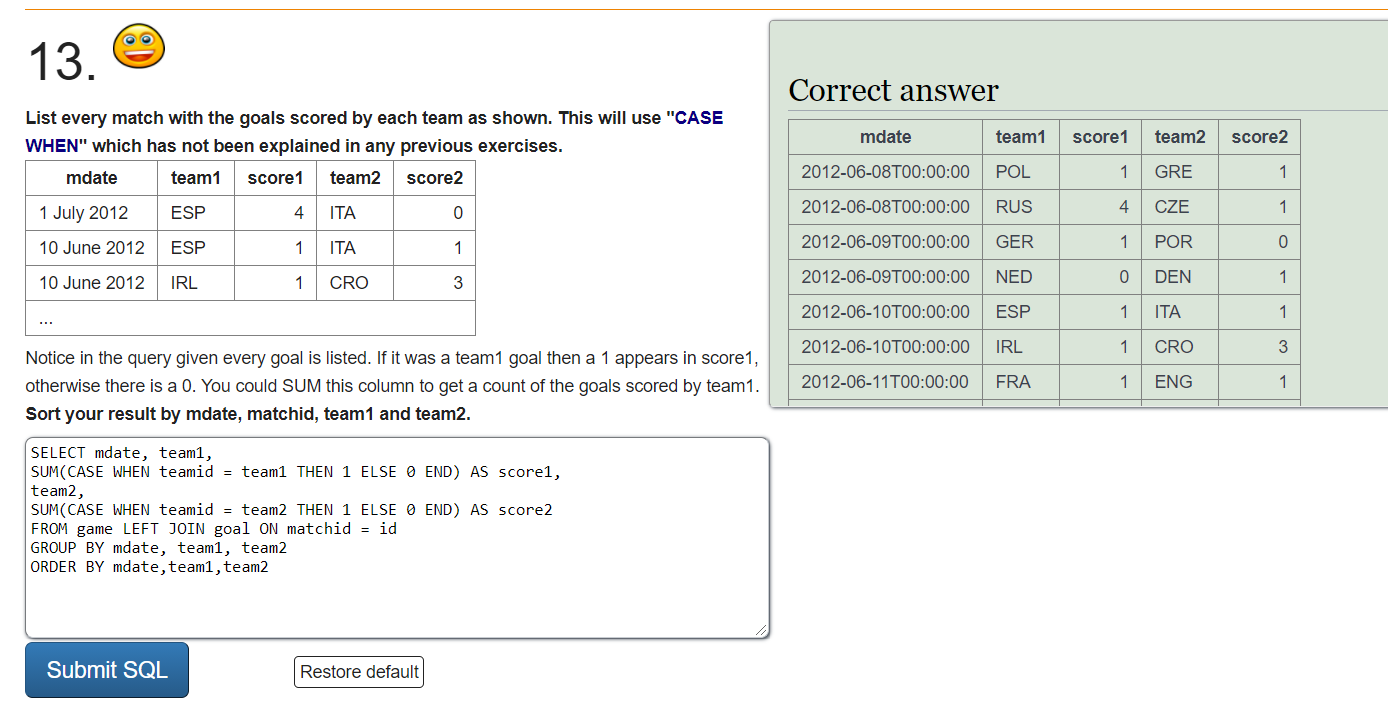












* Создать базу данных university\_db с помощью SQL запроса

**CREATE DATABASE university\_db;**

* Создать сущности:
  + Student
    - student\_id - первичный ключ, автоинкремент.
    - first\_name - имя, varchar(50), Not Null
    - last\_name - фамилия, varchar(50), Not Null
    - data\_of\_birth - дата рождения

**Create table Student(**

**student\_id int auto\_increment,**

**first\_name varchar(50) NOT NULL,**

**last\_name varchar(50) NOT NULL,**

**data\_of\_birth date,**

**primary key (student\_id));**

* + Module
    - module\_id - первичный ключ, автоинкремент.
    - module\_title - название модуля varchar(50), Not Null
    - level - уровень, число, Not Null
    - credits - число, по умолчанию значение 20, Not Null

**Create table module(**

**module\_id int AUTO\_INCREMENT,**

**module\_title varchar(50),**

**level int NOT NULL,**

**credits int default 20,**

**Primary key (module\_id));**

* Каждый студент изучает множество “модулей”. Каждый модуль проходят множество студентов:
  + registration - таблица содержит id студента, id модуля и результат
    - Student\_id - внешний ключ таблицы student
    - module\_id  - внешний ключ таблицы module
    - result - число с одним символом после запятой

**Create table registration (**

**student\_id int,**

**module\_id int,**

**FOREIGN KEY (student\_id) REFERENCES student(student\_id),**

**FOREIGN KEY (module\_id) REFERENCES module(module\_id));**

* Добавить данные для таблиц:
  + student - [Daniel Radcliffe, 1995-04-23], [Emma Watson, 1994-07-15], [Rupert Grint, 1995-10-24]

**insert into student (first\_name, last\_name, data\_of\_birth) values ('Daniel', 'Radcliffe', '1995-04-23');**

**insert into student (first\_name, last\_name, data\_of\_birth) values ('Emma', 'Watson', '1994-07-15');**

**insert into student (first\_name, last\_name, data\_of\_birth) values ('Rupert', 'Grint', '1995-10-24');**

* + module - [Math, 1, 30], [Physics, 1], [Chemistry, 1, 15]

**insert into module (module\_title, level, credits) values ('Math', 1, 30);**

**insert into module (module\_title, level) values ('Physics', 1);**

**insert into module (module\_title, level, credits) values ('Chemistry', 1, 15);**

* + registration
    - Daniel Radcliffe - Math, Daniel Radcliffe - Physics, Daniel Radcliffe - Chemistry
    - Emma Watson - Chemistry, Emma Watson - Math
    - Rupert Grint - Physics, Rupert Grint - Math

**insert into registration (student\_id, module\_id) values (1,1);**

**insert into registration (student\_id, module\_id) values (1,2);**

**insert into registration (student\_id, module\_id) values (1,3);**

**insert into registration (student\_id, module\_id) values (2,3);**

**insert into registration (student\_id, module\_id) values (2,1);**

**insert into registration (student\_id, module\_id) values (3,2);**

**insert into registration (student\_id, module\_id) values (3,1);**