**Autoestudio 01 - MBDA**

Jefer Alexis González Romero  
Angel Nicolas Cuervo Naranjo

**INVESTIGACION**

Para que tener conocimiento suficiente sobre la tecnología que vamos a usar en el laboratorio, investiguen las siguientes preguntas. No olviden incluir la bibliografía.

**A.** SQL

* ¿Qué es? ¿Para qué sirve?

Es un lenguaje de programación basado en consultas estructuradas y muy utilizado por gerentes ed empresas y expertos de marketing por su utilidad para manejar bases de datos relacionales y otras operaciones. Literalmente traducen lenguaje de consulta estructurado (SQL), por us siglas en inglés. En este caso, se utilizan bases de datos relacionales y sirve para tomar desicoines empreariales al analizar grandes cantidades de datos y escribir scripts de integración de datos, básicamente. Estas consultas, se logran al modificar estructuras de bases de datos, agregar, actualizar y suprimir datos y, recuperar la información.

* ¿Qué es DML, DLL, DCL, TCL?

Data Manipulation Language (**DML**) Traduce Lenguaje de Manipulación de Datos y permite modificar los datos de una base de datos. Cuenta con elementos Select (Para consultar edterminados registros), Insert (Carga de datos), Update (Modificar valores y registros especificados) y Delete (Eliminación de registros de una tabla de base de datos).

Dinamic Link Libraries (**DLL**) es la parte de SQL encargada a la definición de las bases de datos, consta de sentencias para definir la estructura de bases de datos.

Data Control Language (**DCL**) El Lenguaje de control ed datos es tilizado para asignación ed roles y el control de acceso. Sus elementos son el Grant (Otorga privilegios de acceso a la DB) y Revoke (Revoa privilegios de aceso con el comando GRANT).

Transactional Control Language (**TCL**) El Lenguaje de Control Transaccional administra transacciones dentro de la base de datos. Utiliza los elementos Commit (Para guardar el trabajo hecho) y Rollback (Deshace la modificación después del commit).

* En este autoestudio, ¿en qué escribimos? ¿por qué?

En DML ya que, a menudo, necesitamos realizar operaciones de Selección, Inserccion, Actualzación y Borrado de datos.

**B.** Motor de bases de datos y bases de datos

* ¿Qué son?

Motor de bases de datos Es el componente del programa informático que permite gestionar la base de datos al crear, leer, actualizar y eliminar datos de una base de datos.

Bases de datos Es un sistema de información que permite interrelacionar datos mediante una red, con el fin de compartir información entre Bases de datos, servidor y cliente.

* ¿Qué motores ofrece sqlzoo.net [<http://sqlzoo.net/>]?

SQL server, MySQL, DB2, Oracle y PostgreSQL

* ¿Qué bases de datos ofrece sqlzoo?

Adventure Works, University Timetables, Musicians, DressMaker, Cogestion Chaning

**Bibliografía:**

DATE C.J. “INTRODUCCIÓN A LOS SISTEMAS DE BASES DE DATOS”. PRENTICE HALL. 2001 SÉPTIMA EDICIÓ https://www.clubensayos.com/Tecnolog%C3%ADa/SQL-y-MOTORES-DE-BASES-DE-DATOS/3806408.html N.

**PRÁCTICA**

**A.** Estudien las secciones SELECT, SELECT ...WHERE, SELECT ...GROUP BY, SELECT ...SELECT de la referencia y escriban expresiones para todas las consultas en cálculo y algebra.

**SELECT**

SQL

SELECT columnas

FROM tabla

Algebra

Πcolumnastabla

Cálculo

{x : tabla |: columnas}

**SELECT ...WHERE**

SQL

SELECT \*

FROM tabla

WHERE condición

Algebra

σcondicióntabla

Cálculo

{x : tabla | condición : x}

**SELECT … SELECT**

SQL

SELECT columna

FROM (SELECT columna FROM TABLA)

Algebra

Πcolumnas(Π columnastabla)

Cálculo

{x : {y : tabla |: columnas} |: columnas}

**B.** Estudien la section FUNCTIONS de la referencia, seleccionen 5 funciones y escriban 5 consultas que usen product de la base de datos Adventure Works

**Función: CONCAT**

SELECT CONCAT(Name,' tiene ID ', ProductModelID, ' y es de color ', Color)

FROM Product

Where Color LIKE 'W%'



**Función: AVG**

SELECT Color, AVG(ListPrice)

FROM Product

GROUP BY Color

**Función: MAX**

SELECT Sze, MAX(ListPrice)

FROM Product

GROUP BY Sze



**Función: SUM**

SELECT Color, SUM(Weight)

FROM Product

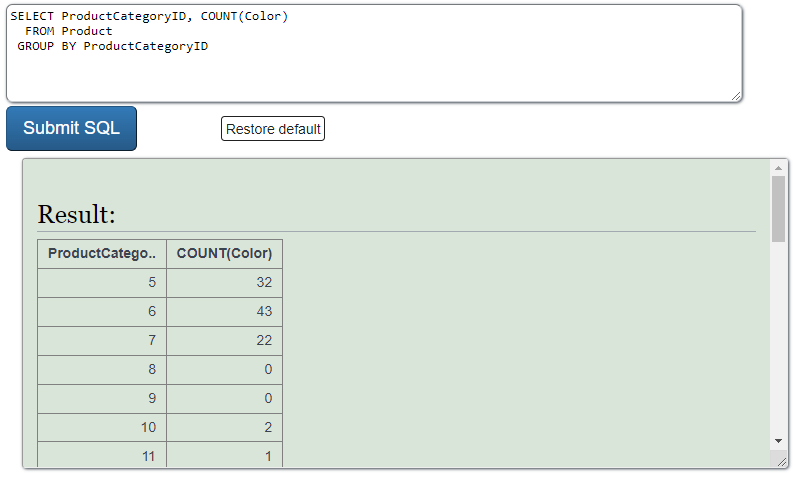
GROUP BY Color

**Función: COUNT**

SELECT ProductCategoryID, SUM(Color)

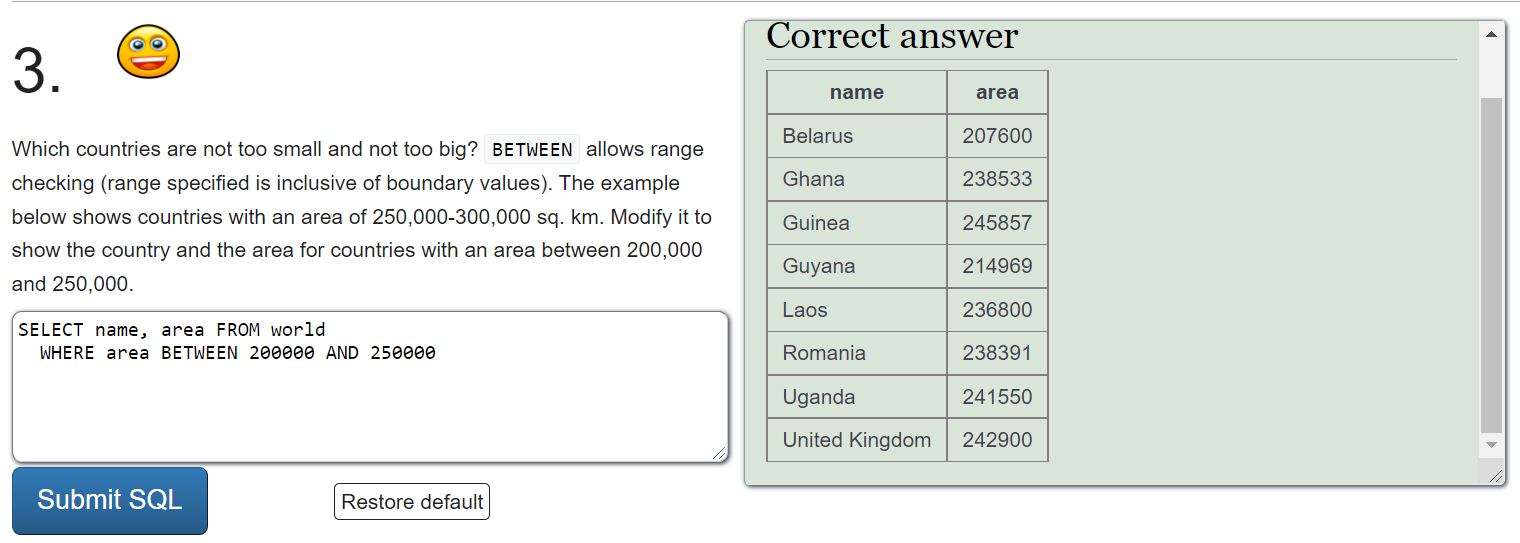
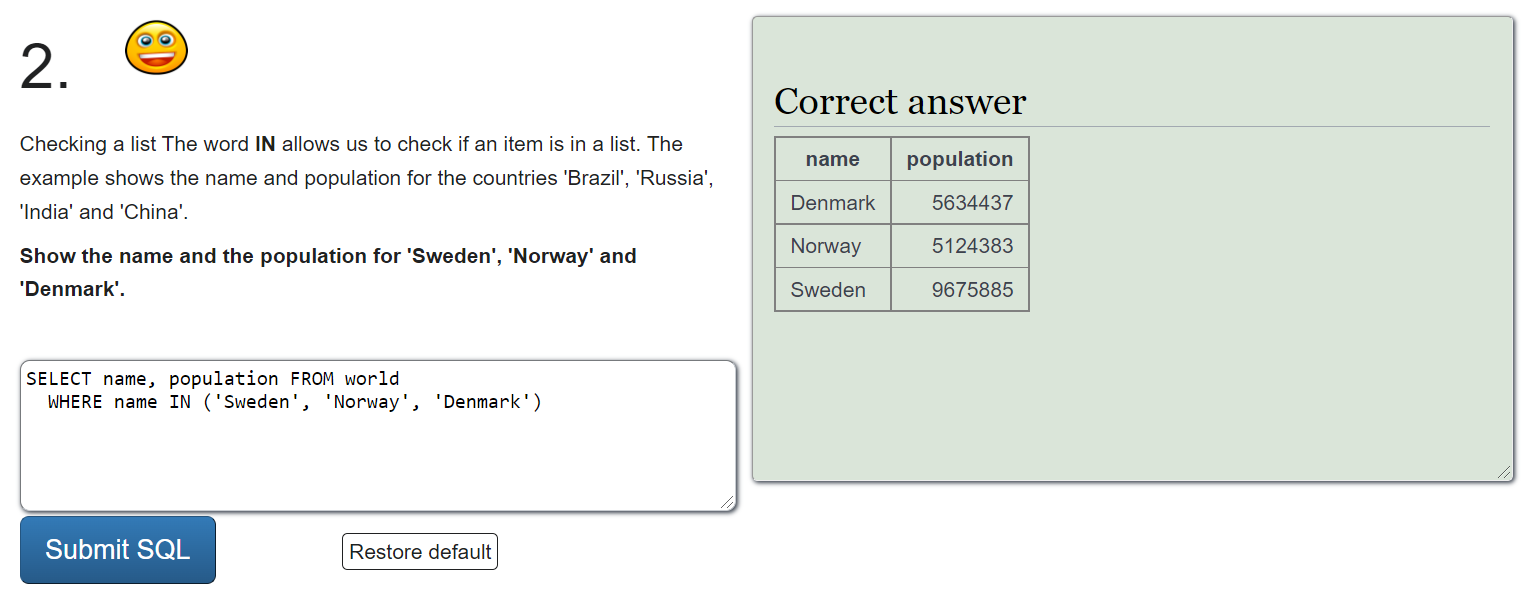
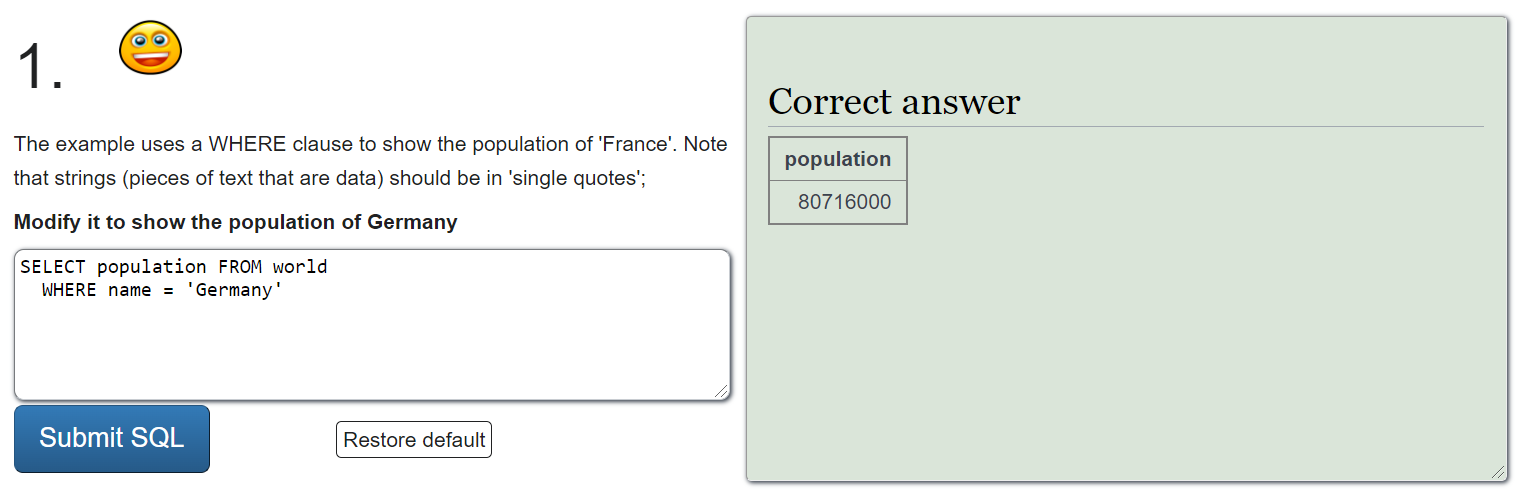
FROM Product

GROUP BY ProductCategoryID

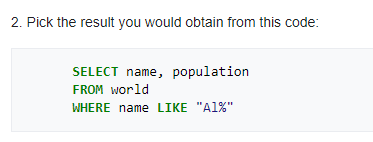
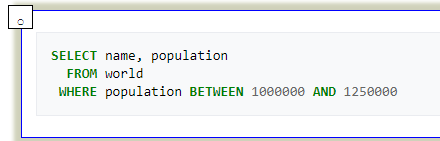
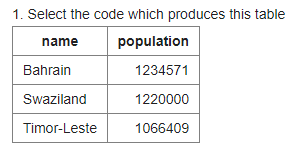


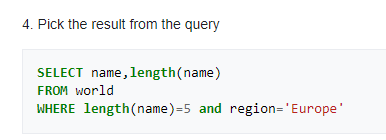
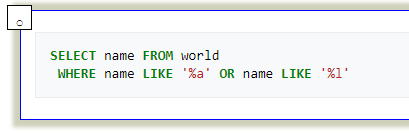
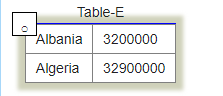
**C.** Realicen todos los ejercicios propuestos en los siguientes tutoriales. Utilice el motor My SQL.

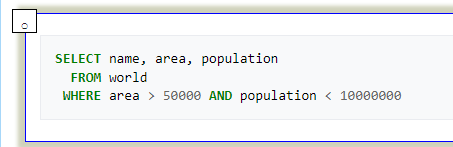
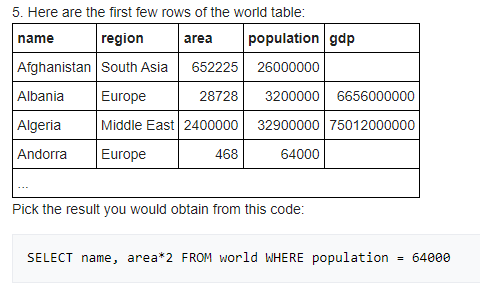
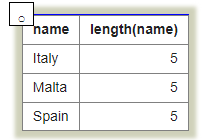
**SELECT basics**



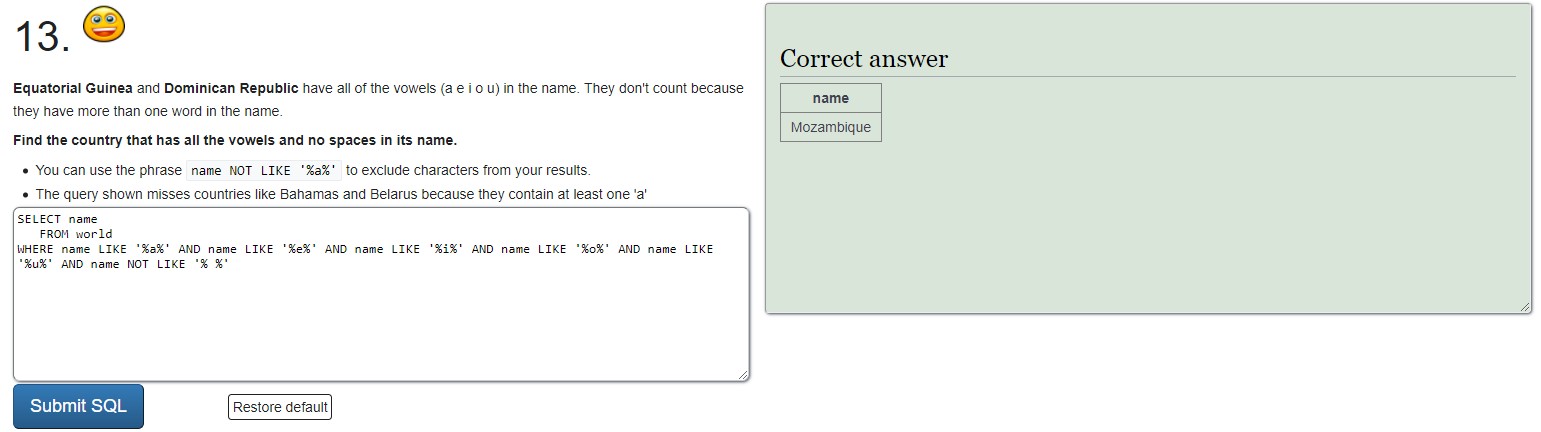
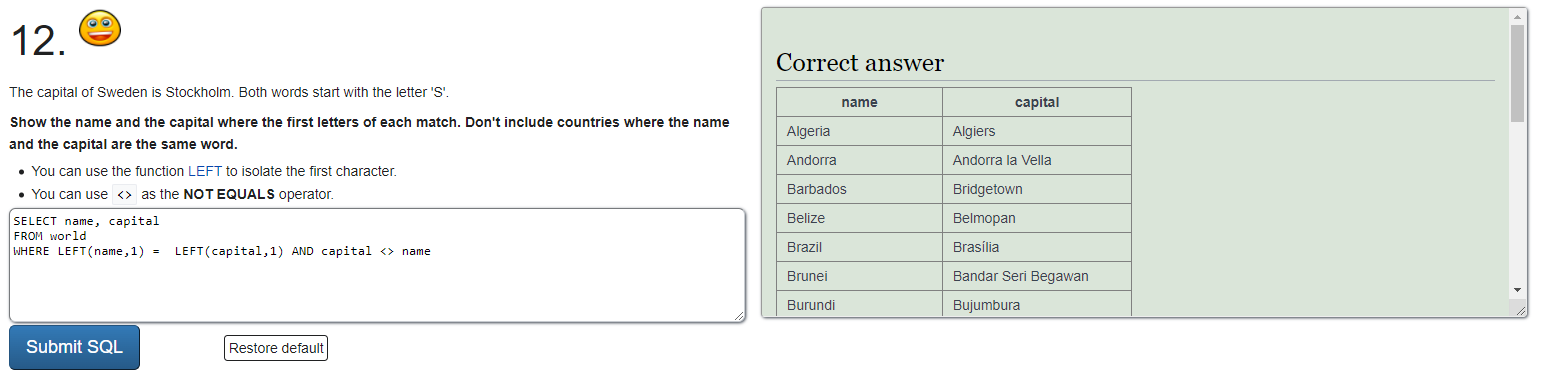
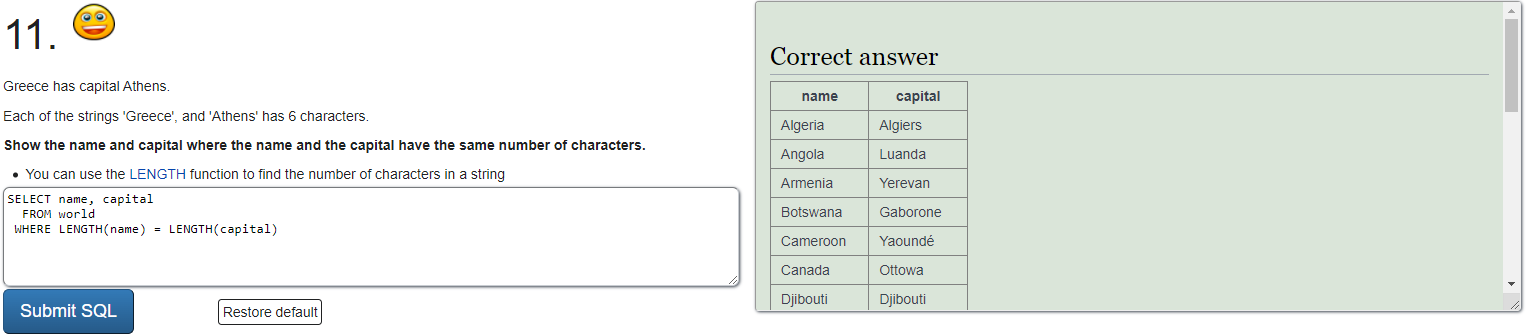
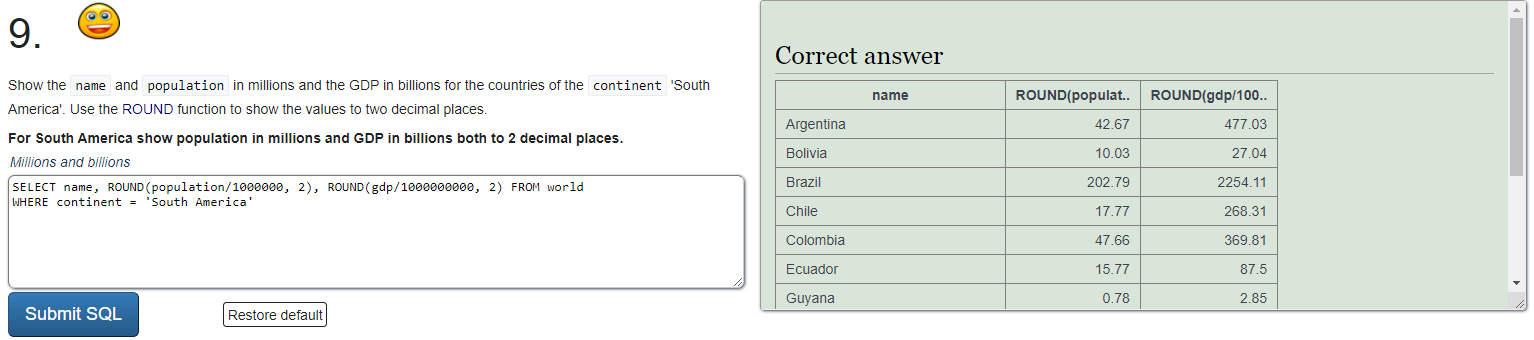
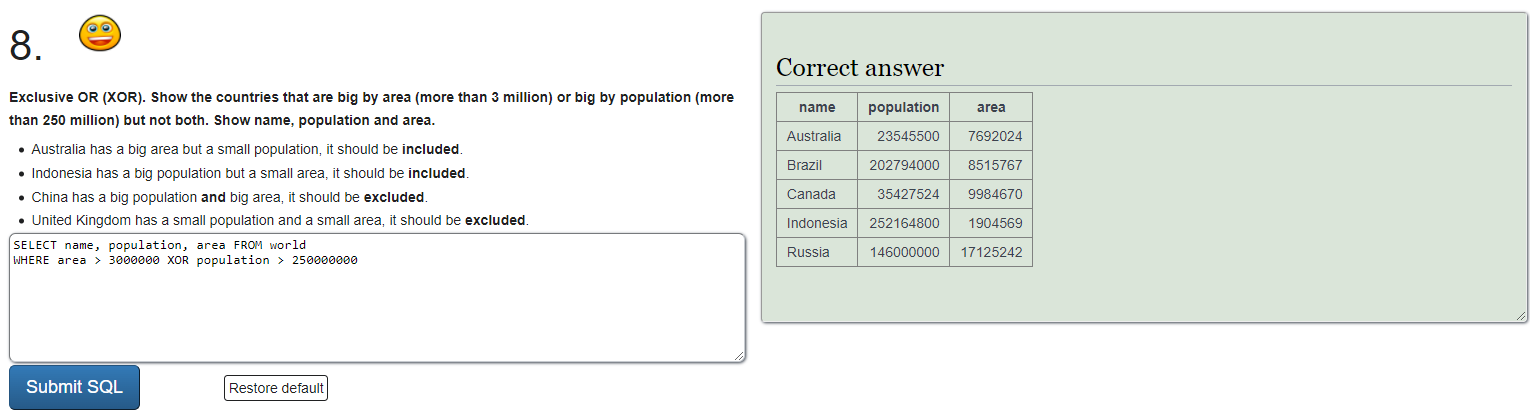
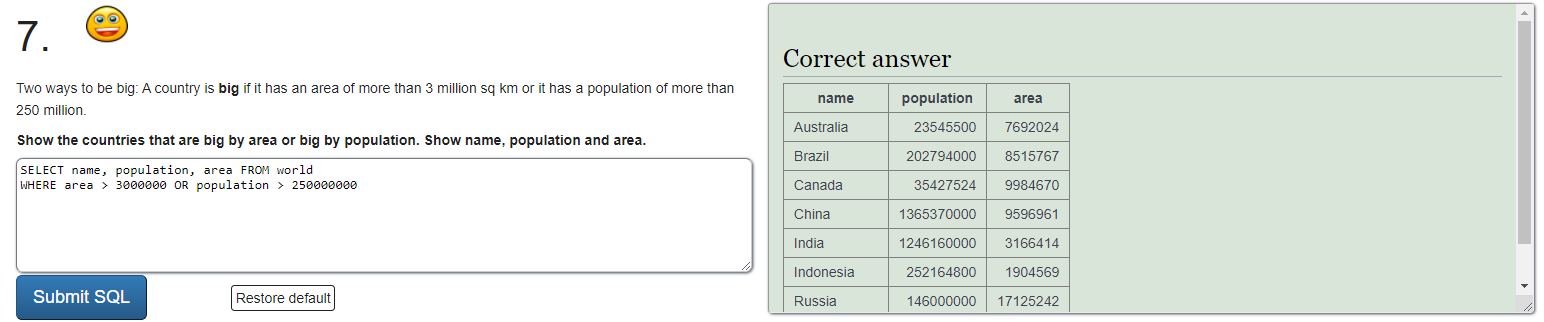
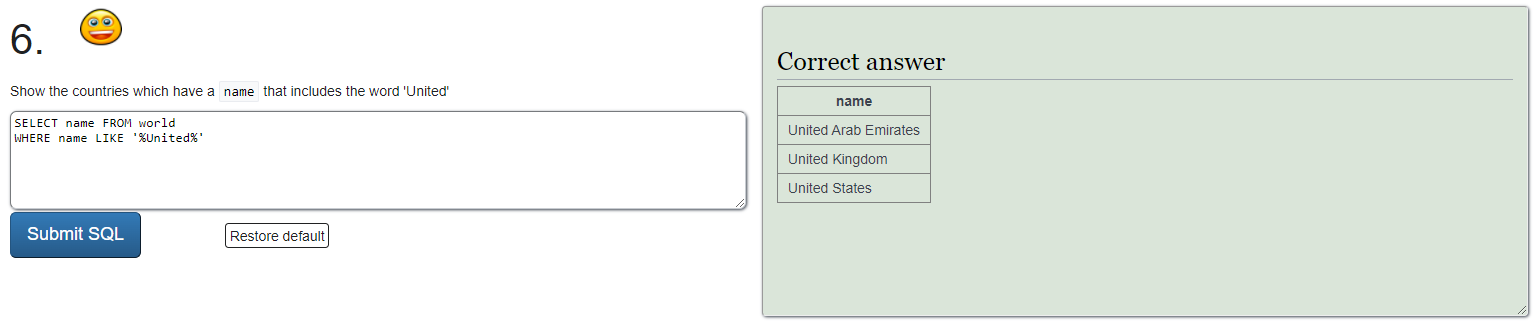
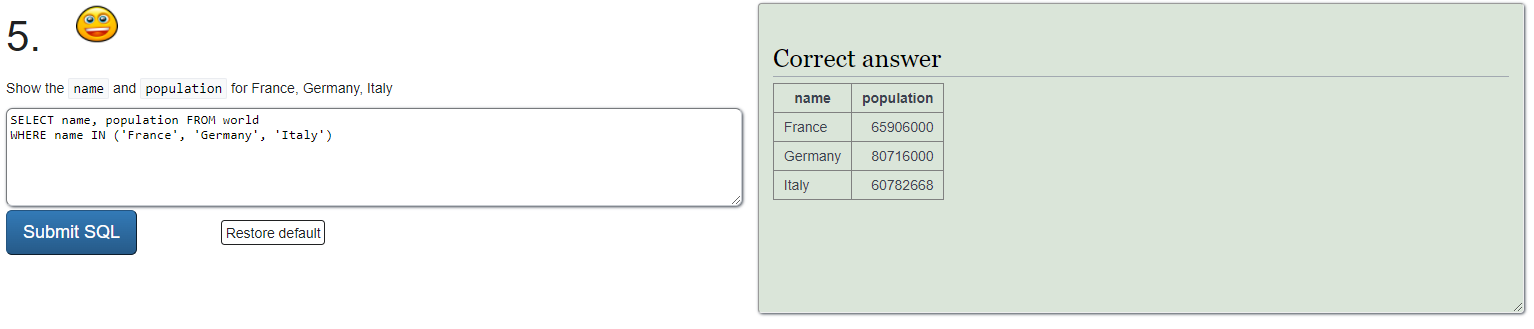
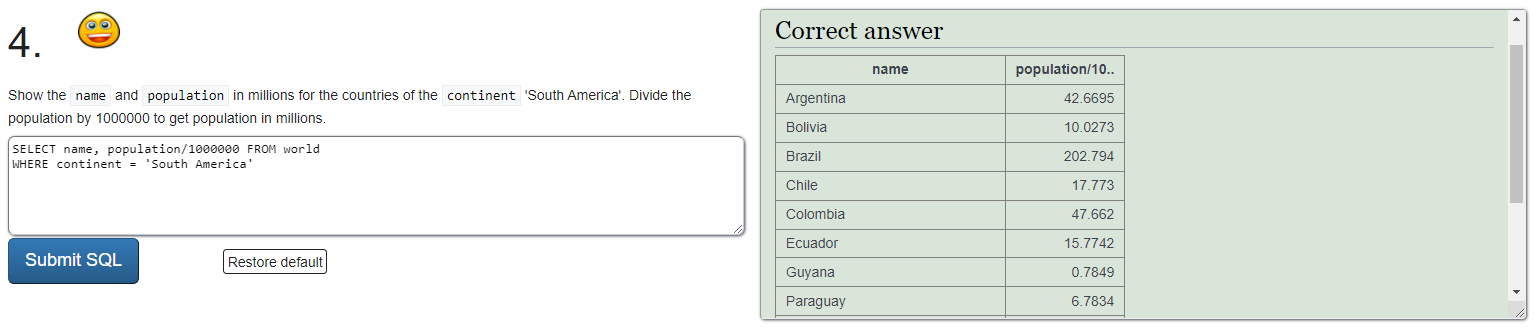
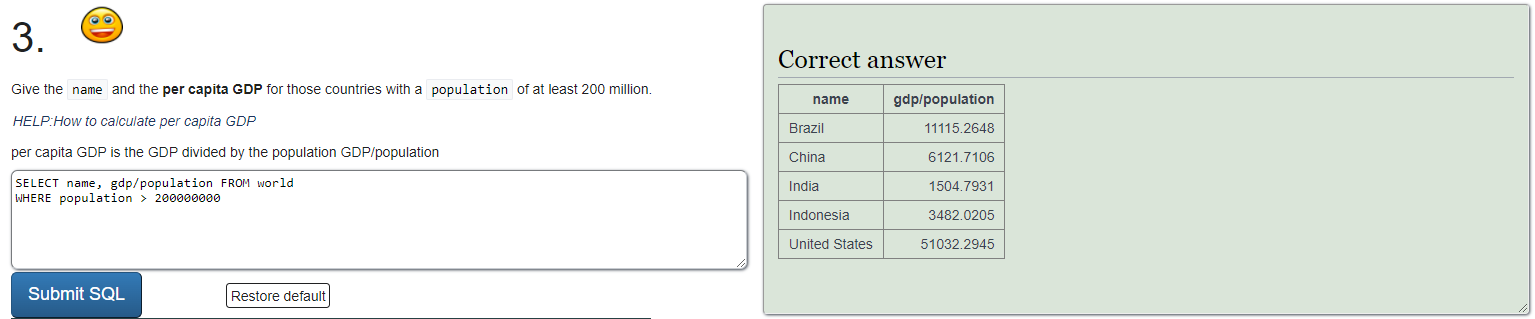
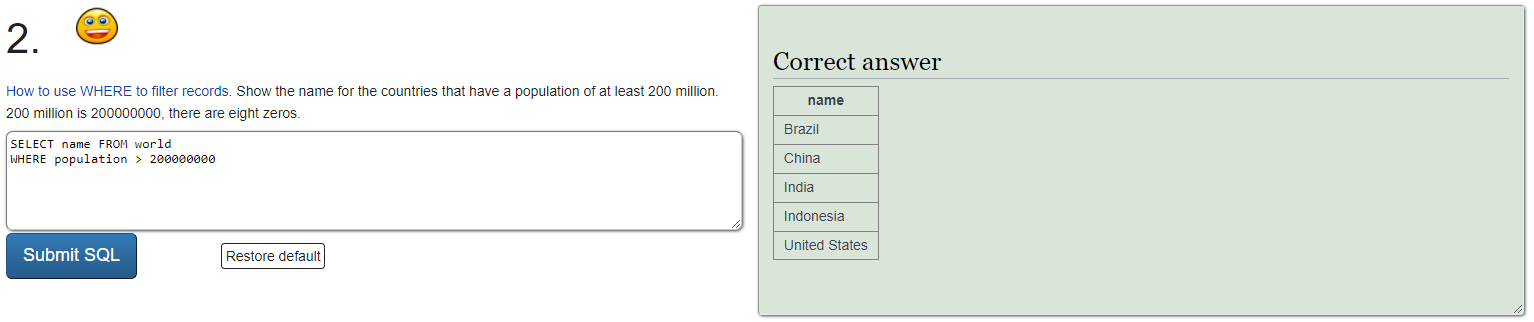
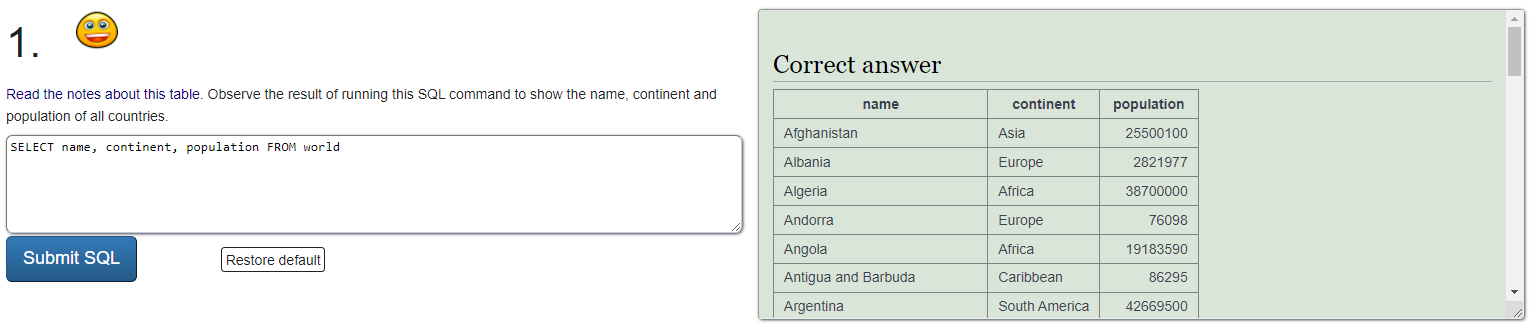
**Quiz**



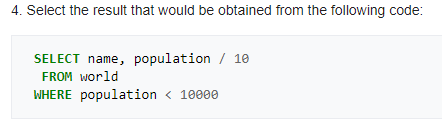
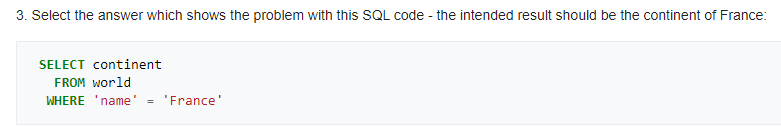
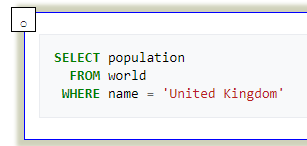
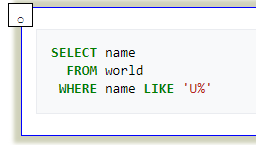


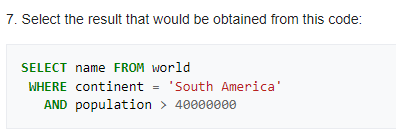
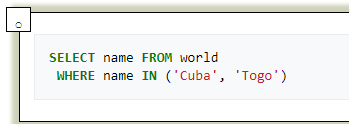
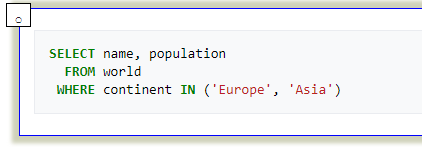


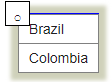
**SELECT from world**



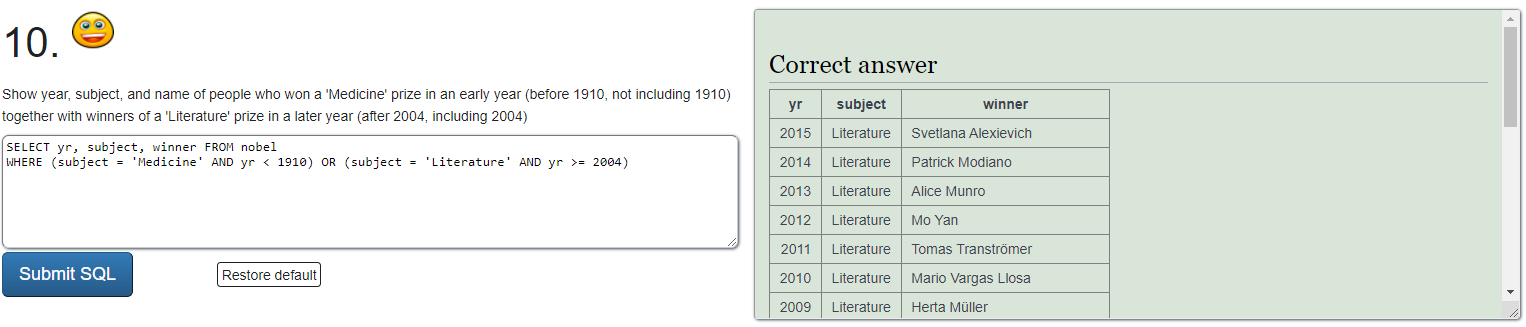
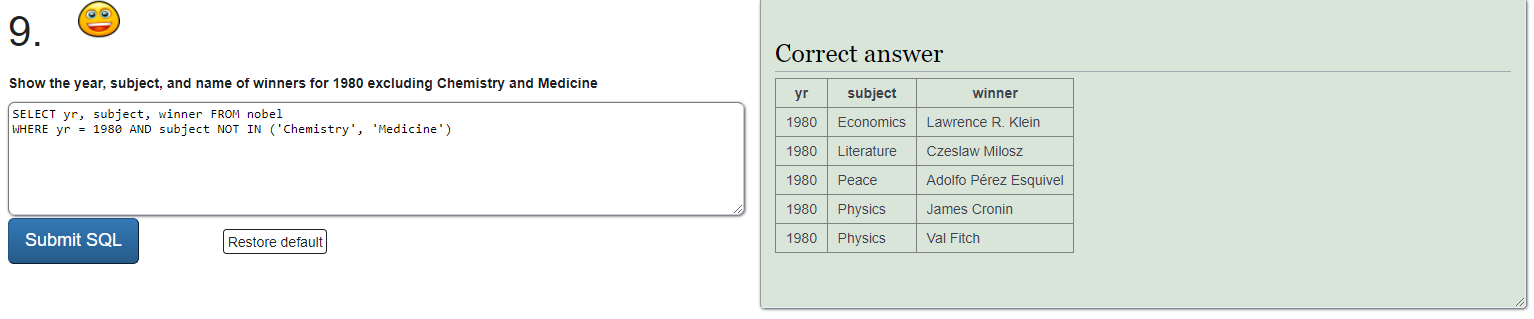
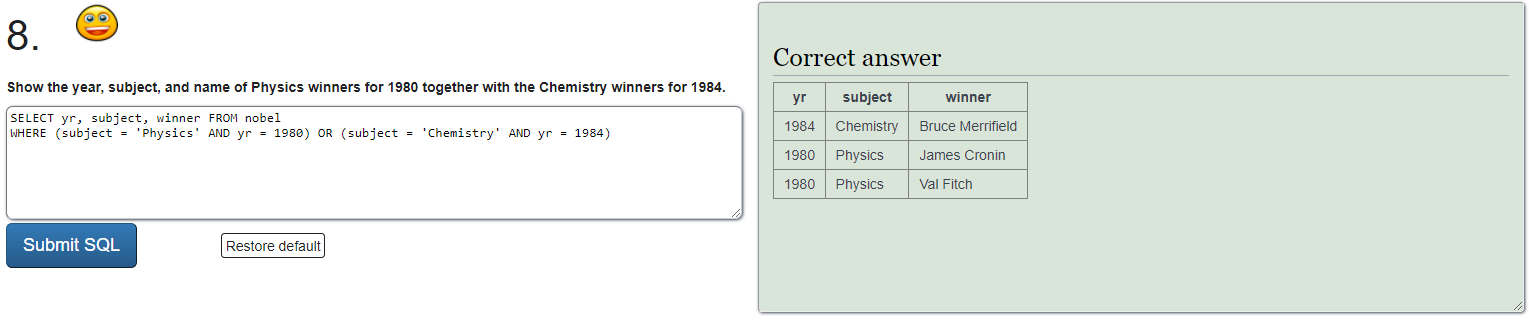
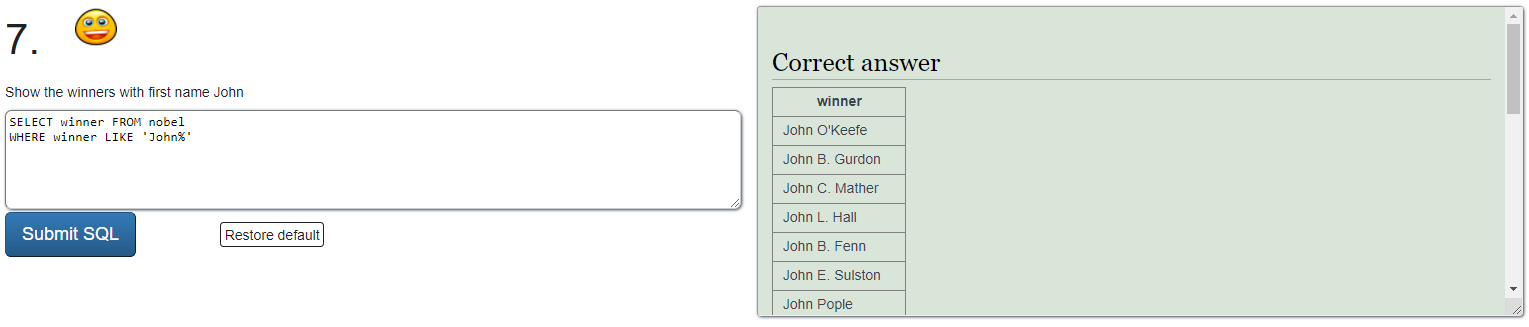
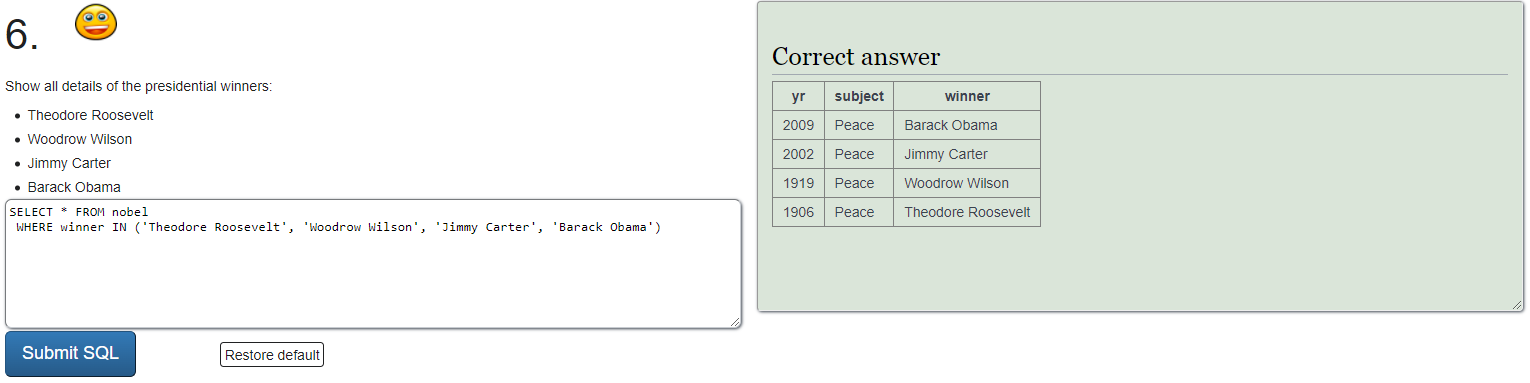
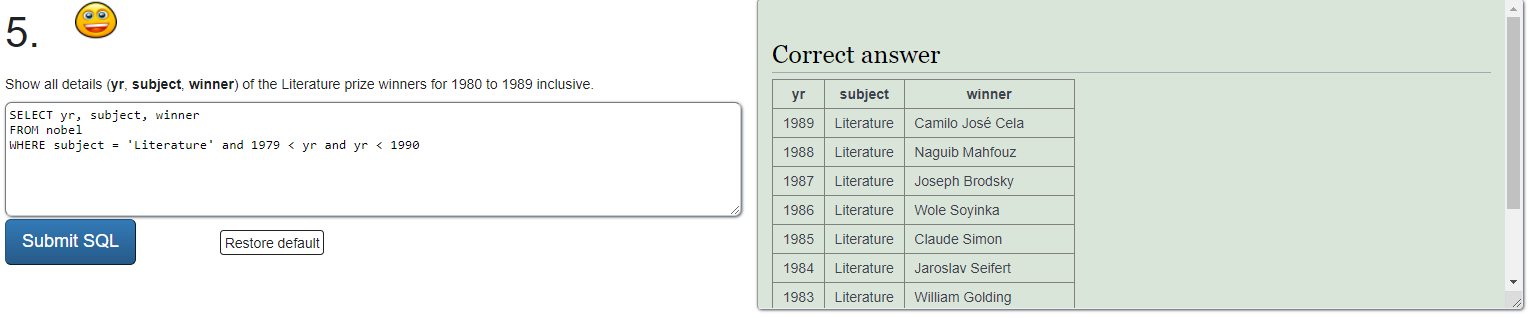
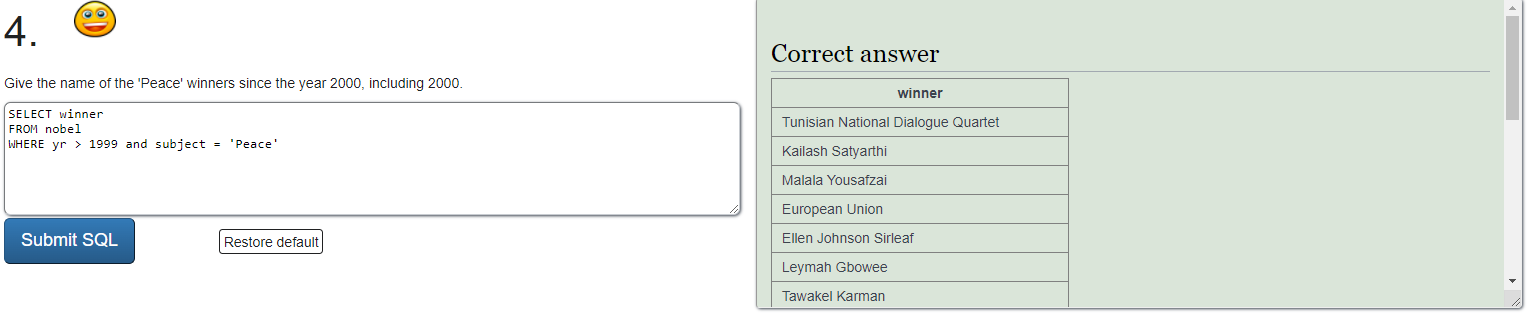
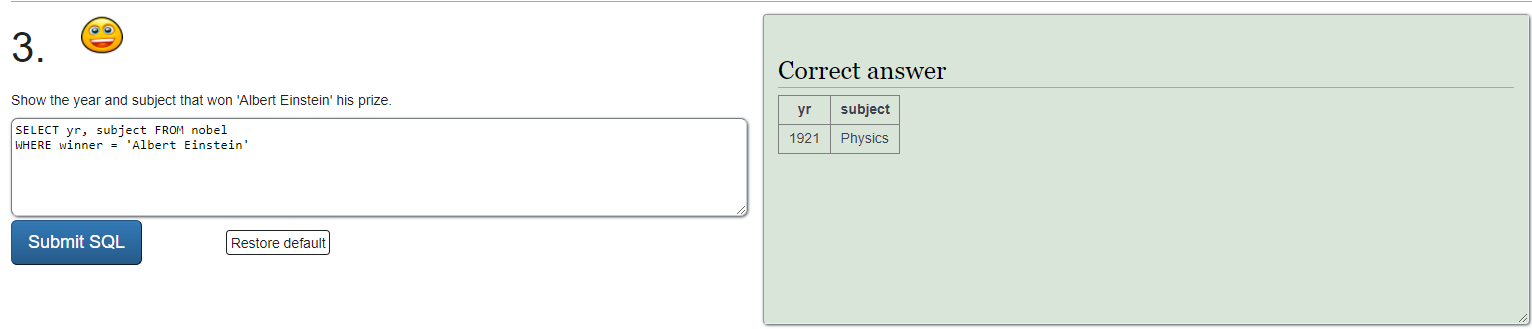
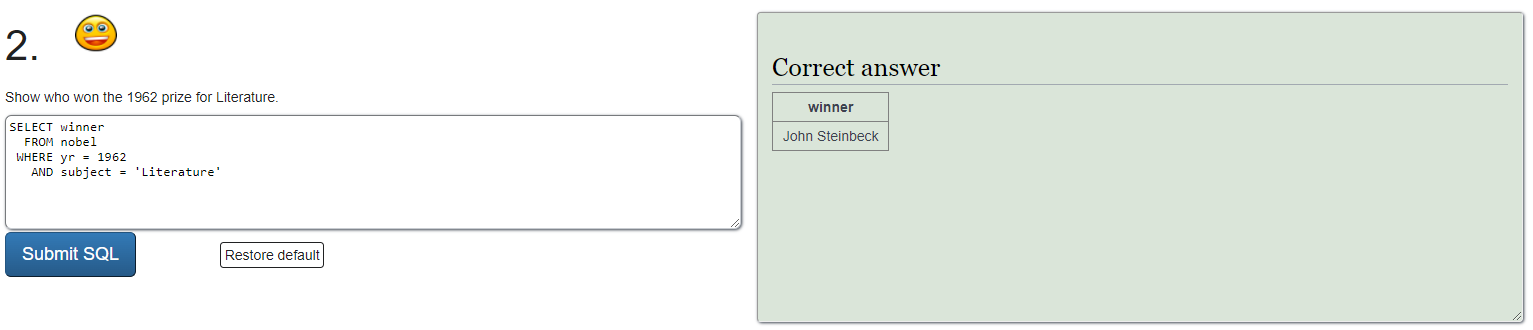
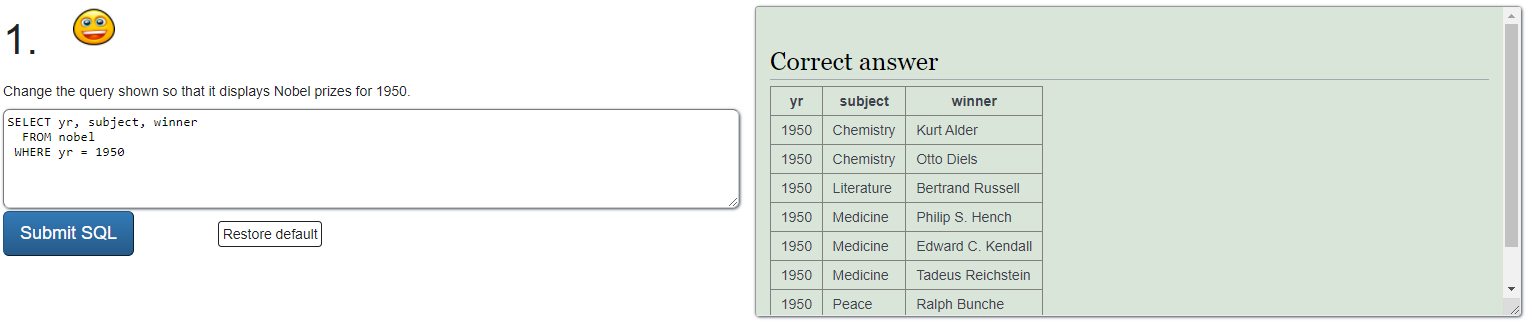
**QUIZ**

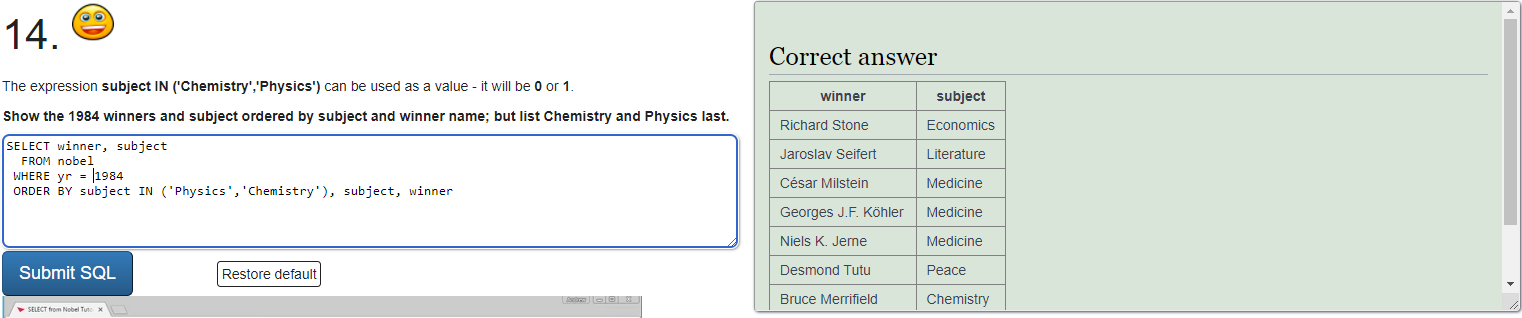
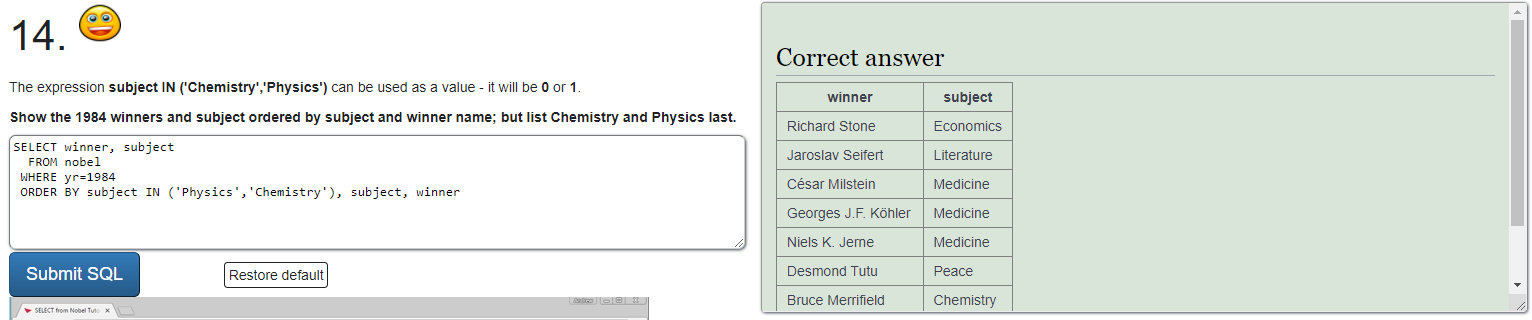
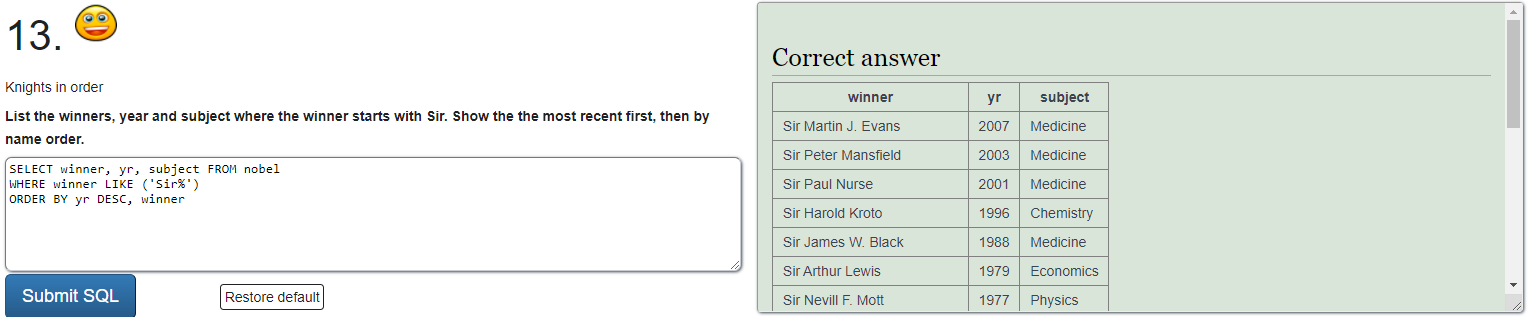
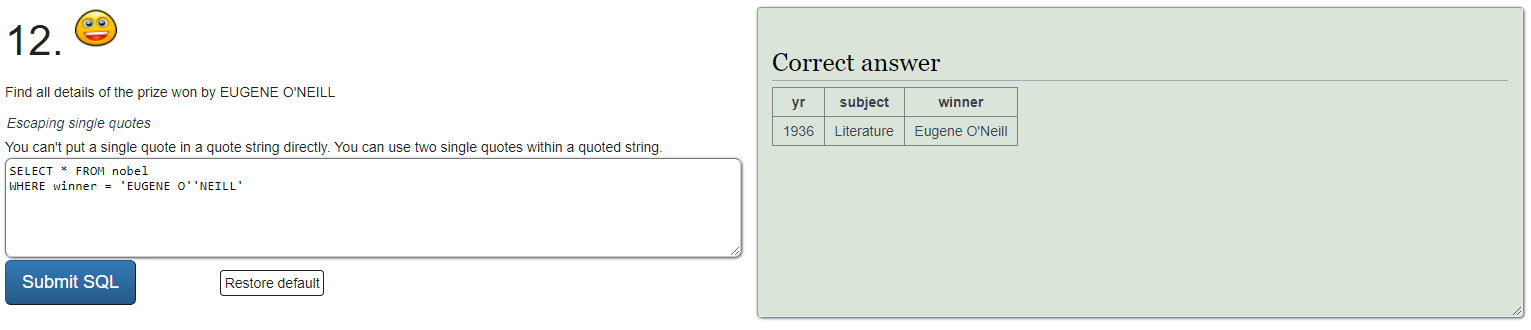
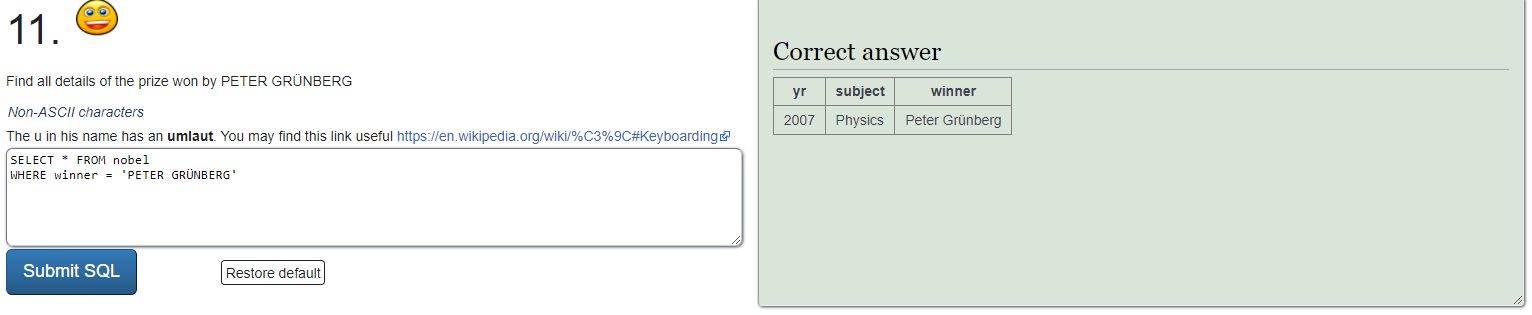




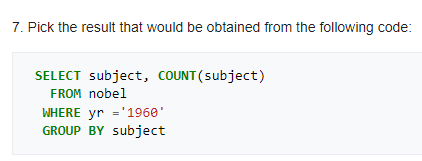
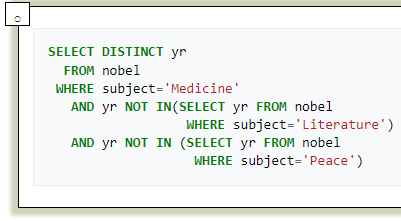
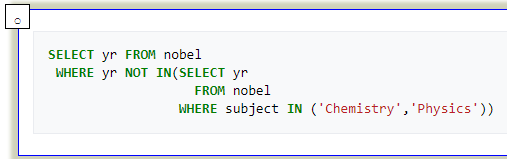
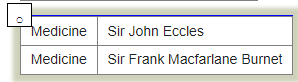
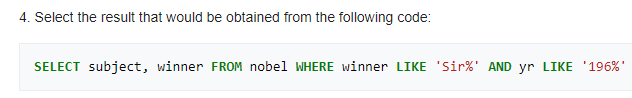
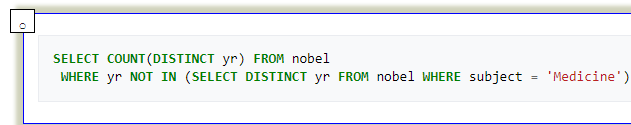
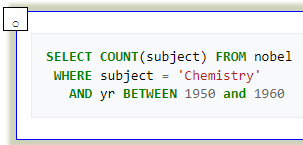
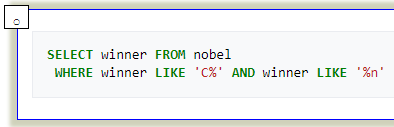


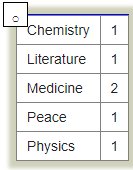
**SELECT from nobel**



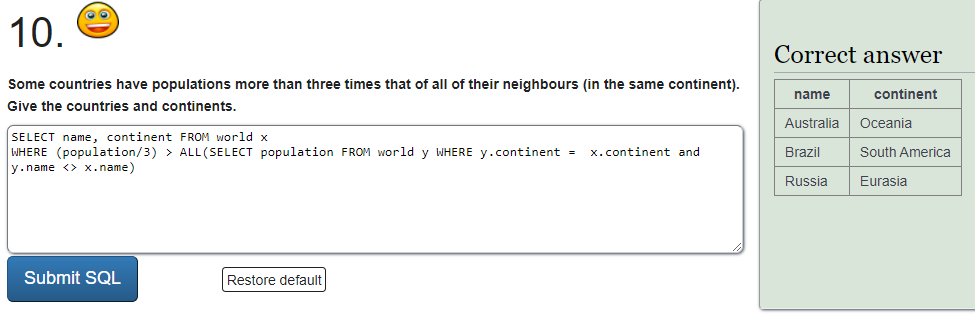
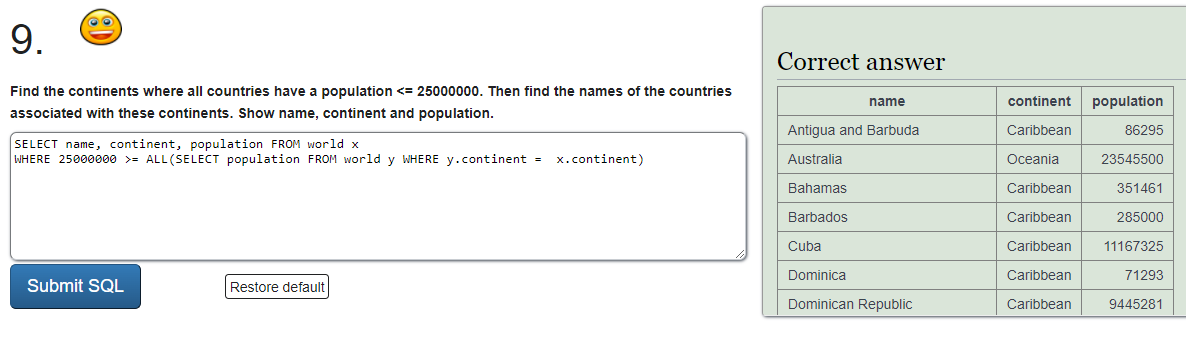
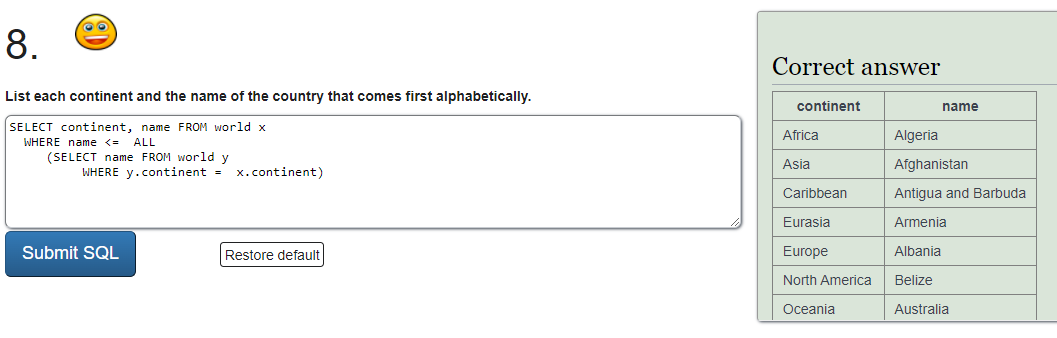
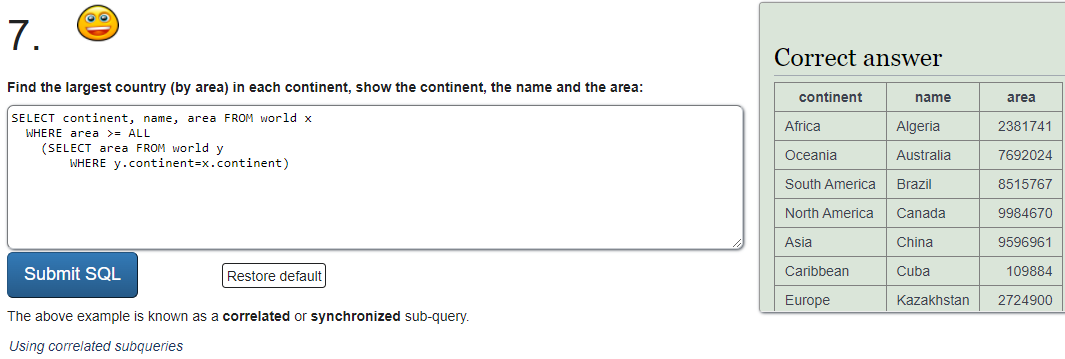
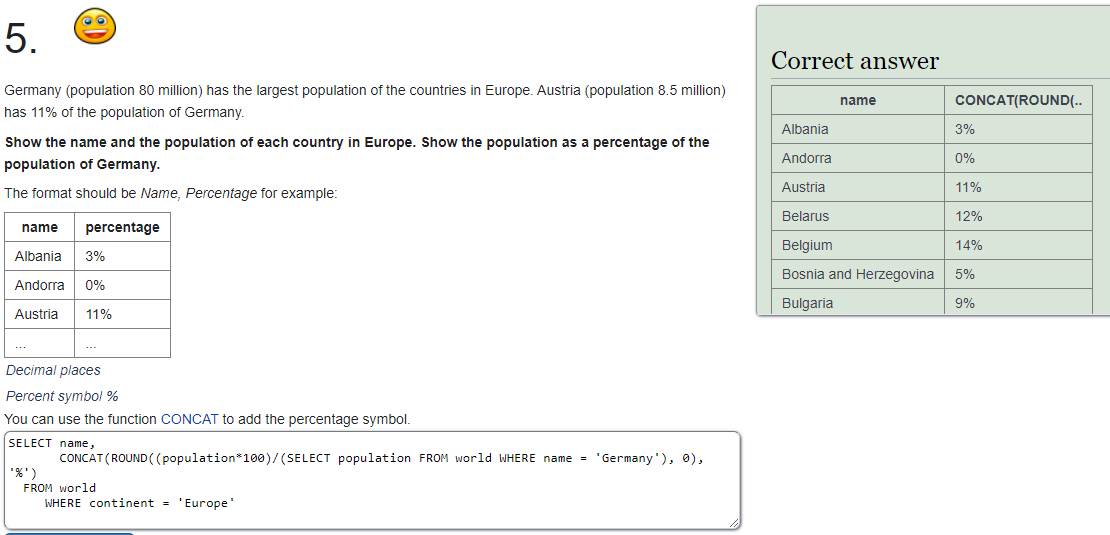
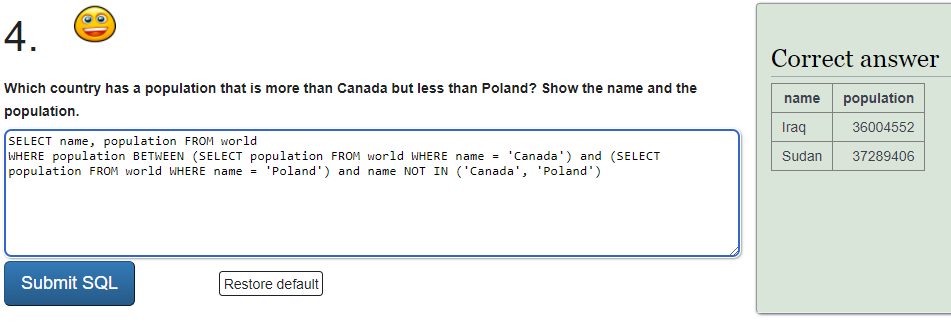
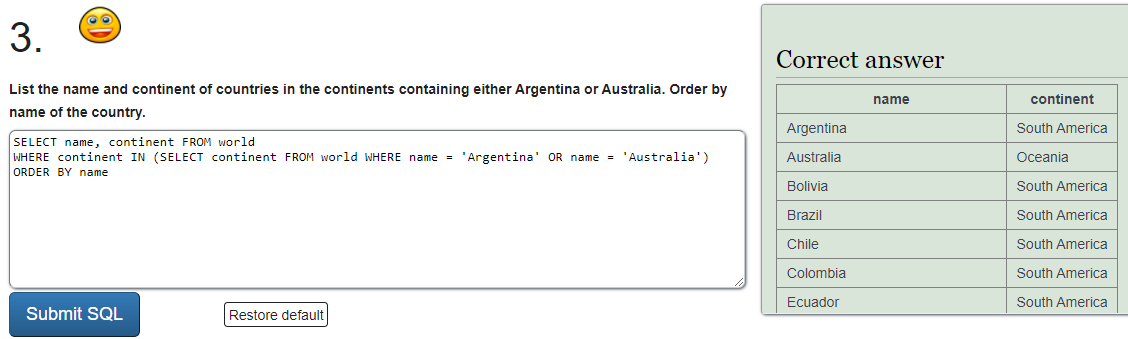
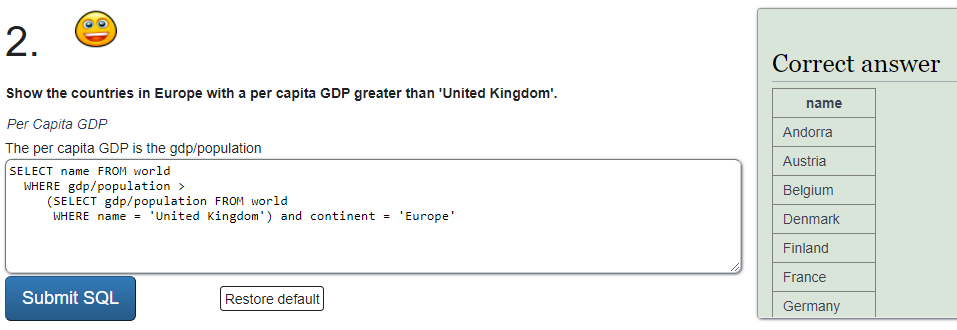
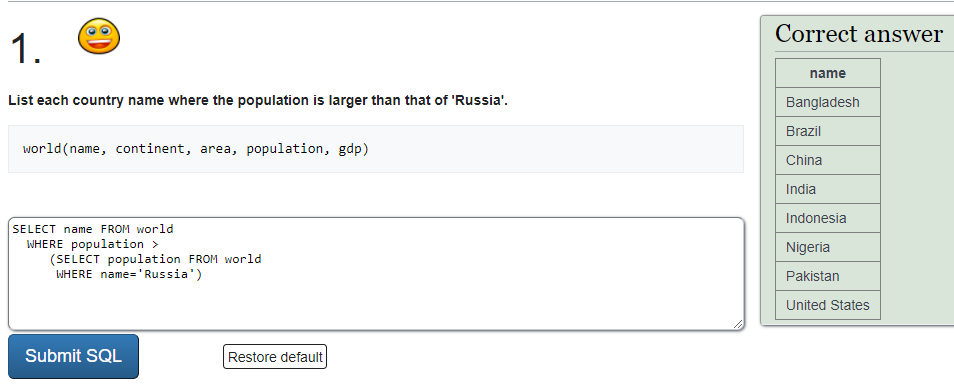


**QUIZ**

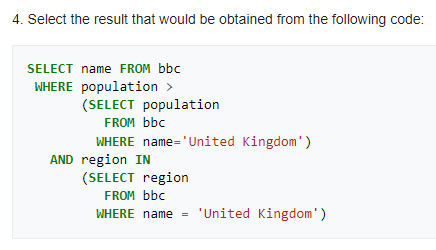
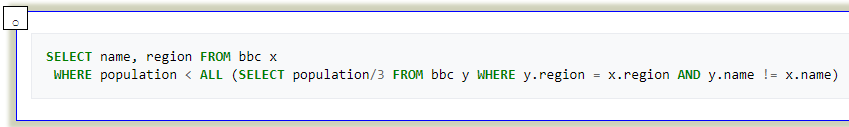
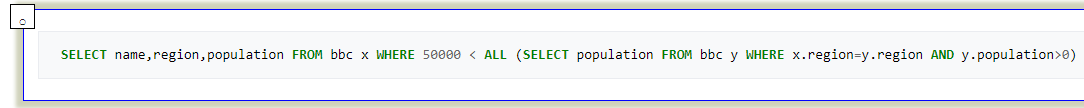
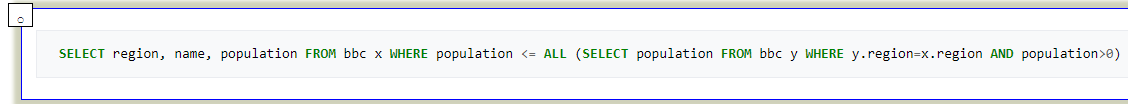




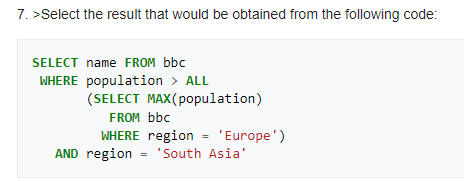
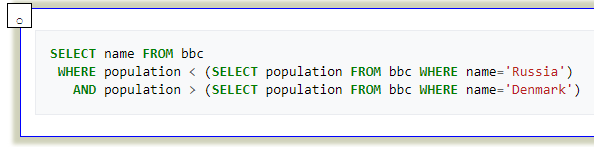
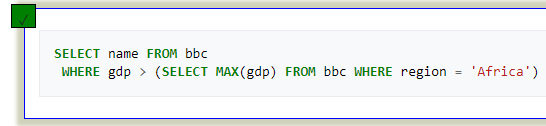
**SELECT in SELECT**

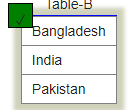


**QUIZ**

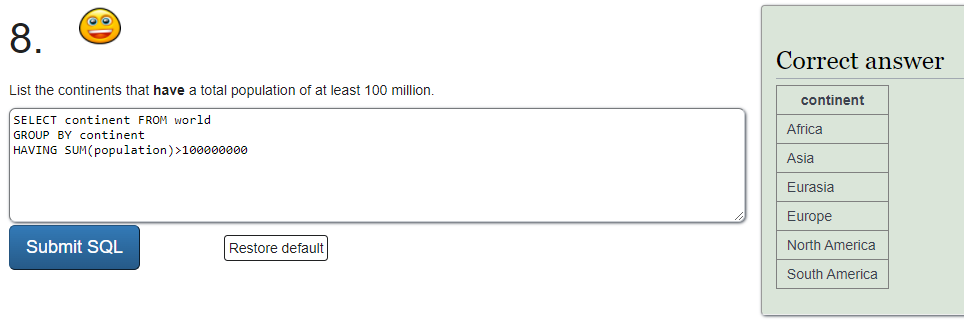
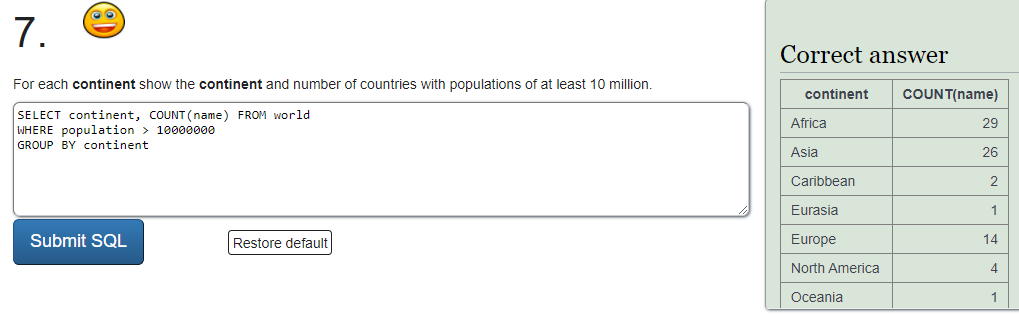
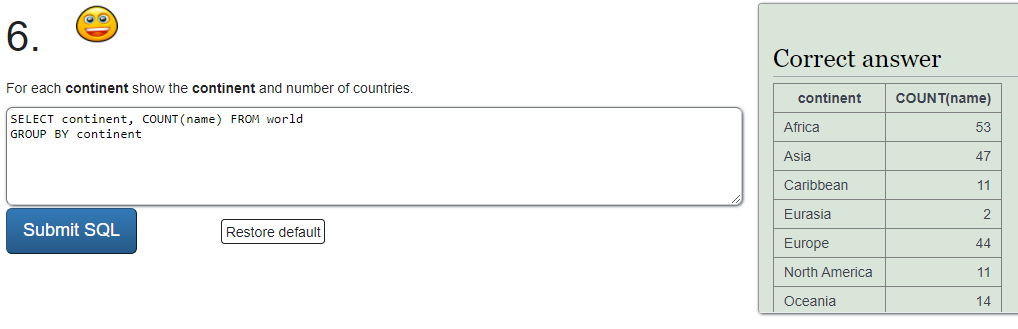
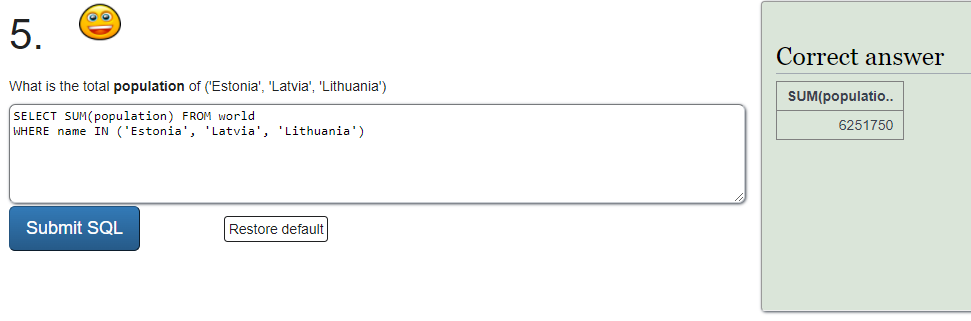
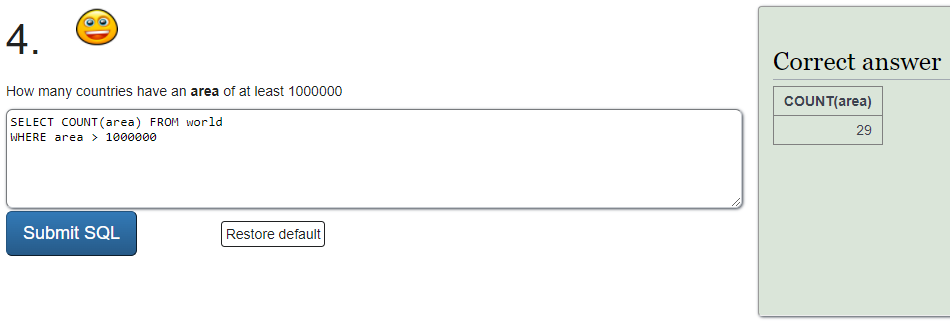
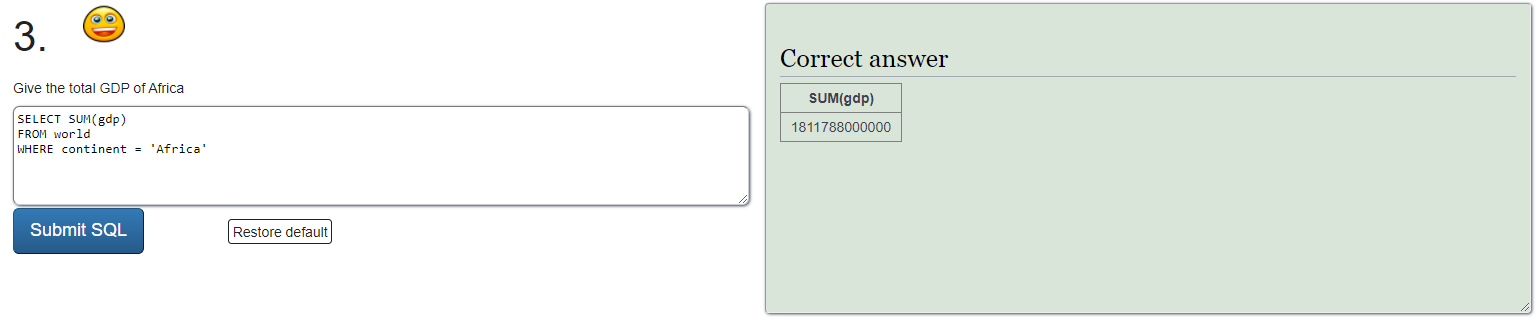
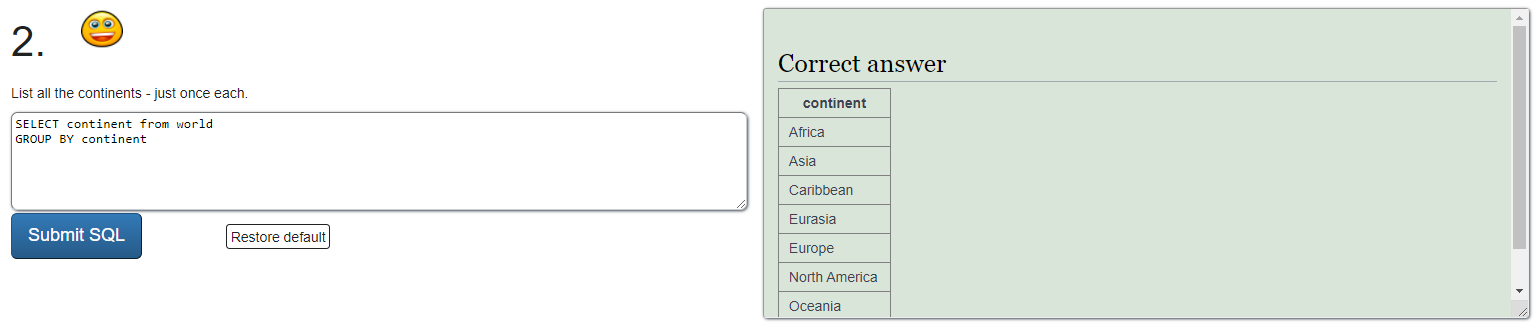
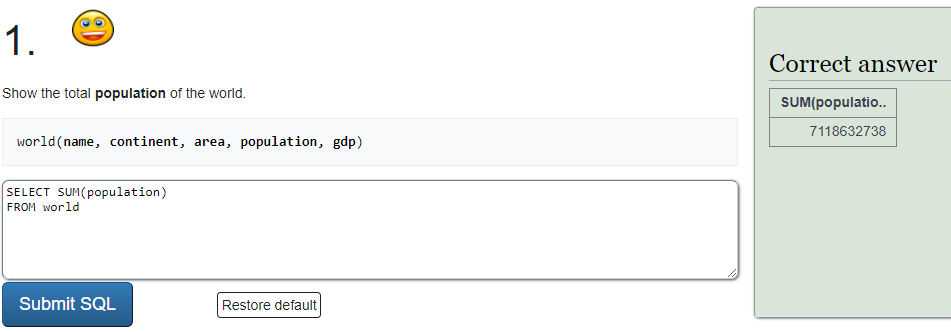




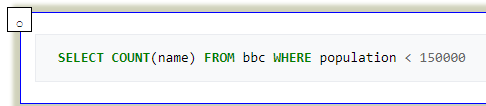
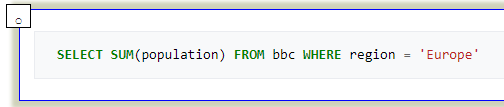




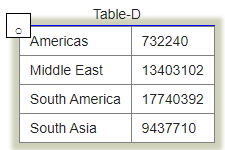
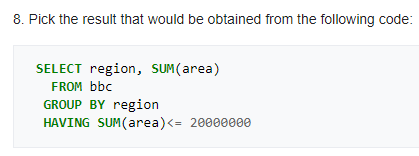
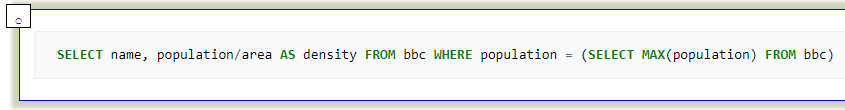
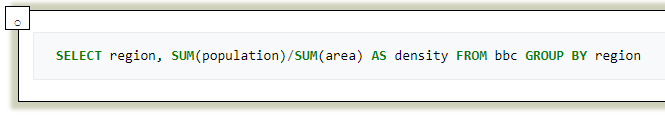
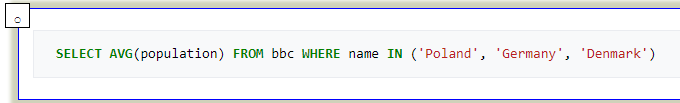
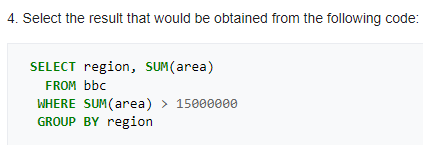
**SELECT and COUNT**



**QUIZ**







**D.** De las consultas anteriores, escriban 5 en algebra y 5 en cálculo.

**ALGEBRA**

Show the name and the population for 'Sweden', 'Norway' and 'Denmark'

Πname, population(σname = ‘Swden’ ∨ name = ‘Norway’ ∨ name = ‘Denmark’ world)

Show the name for the countries that have a population of at least 200 million.

Πname(σpopulation > 200000000world)

Show the name and population for France, Germany, Italy

Πname, population(σname = ‘France’ ∨ name = ‘Germany’ ∨ name = ‘Italy’ world)

Show the countries that are big by area or big by population. Show name, population and area. (A country is **big** if it has an area of more than 3 million sq km or it has a population of more than 250 million.)

Πname, population, area(σarea > 3000000 ∨ population > 250000000 world)

Show who won the 1962 prize for Literature.

Πwinner(σyr = 1962 ∧ subject = ‘Literature’nobel)

**CÁLCULO**

Show the year and subject that won 'Albert Einstein' his prize.

{y : {x : nobel | winner = ‘Albert Einstein’} | : yr ∧ subject}

Give the name of the 'Peace' winners since the year 2000, including 2000.

{y : {x : nobel | yr >= 2000 and subjec = ‘Peace’} | : winner}

Show all details (**yr**, **subject**, **winner**) of the Literature prize winners for 1980 to 1989 inclusive.

{y : {x : nobel | 1980 <= yr <= 1989 and subject = ‘Literature’ } | : yr ∧ subject ∧ winner}

Show the year, subject, and name of Physics winners for 1980 together with the Chemistry winners for 1984.

{y : {x : nobel | (subject = ‘Physics’ and yr = 1980) ∨ (subject = ‘Chemistry’ and yr = 1984)} | : yr ∧ subject ∧ winner}

Show the year, subject, and name of winners for 1980 excluding Chemistry and Medicine

{y : {x : nobel | yr = 1980 and subject ≠ ‘Chemistry’ and subject ≠ ‘Medicine’} | : yr ∧ subject ∧ winner}

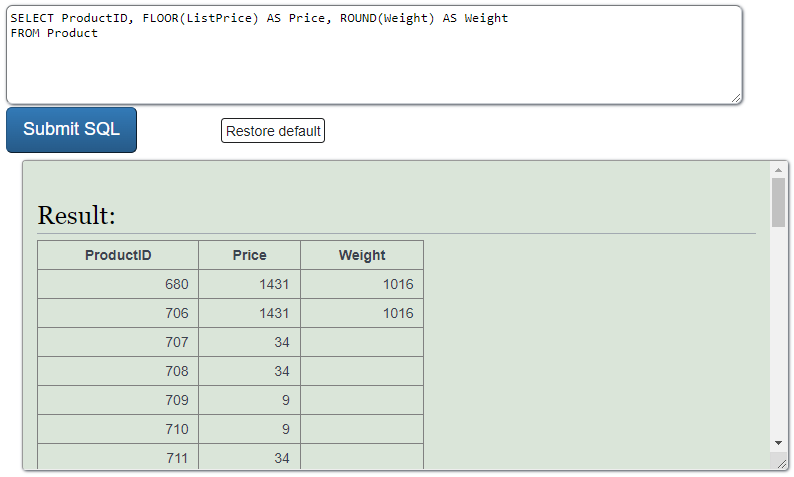
**E.** Propongan consultas que cumplan los siguientes requerimientos. Usen product de la base de datos Adventure Works.

* 8 consultas: una para cada uno de los tipos de operadores.

Numéricos:

SELECT ProductID, **FLOOR**(ListPrice) AS Price, **ROUND**(Weight) AS Weight

FROM Product

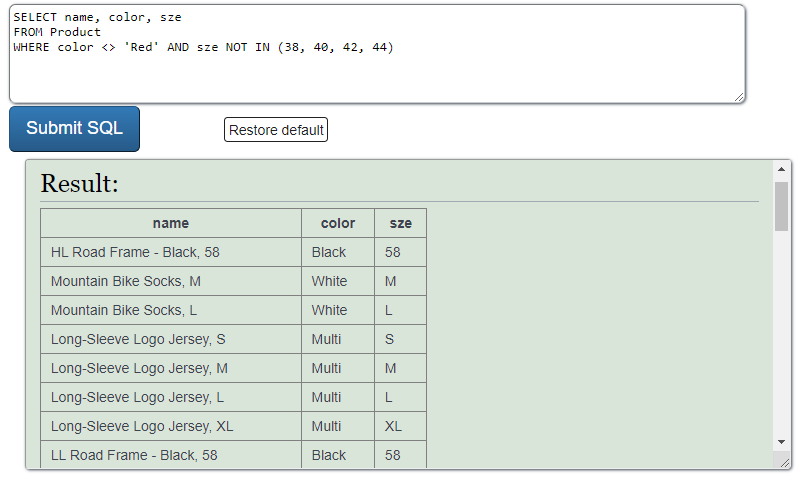


Lógicos:

SELECT name, color, sze

FROM Product

WHERE color <> 'Red' **AND** sze **NOT** IN (38, 40, 42, 44)

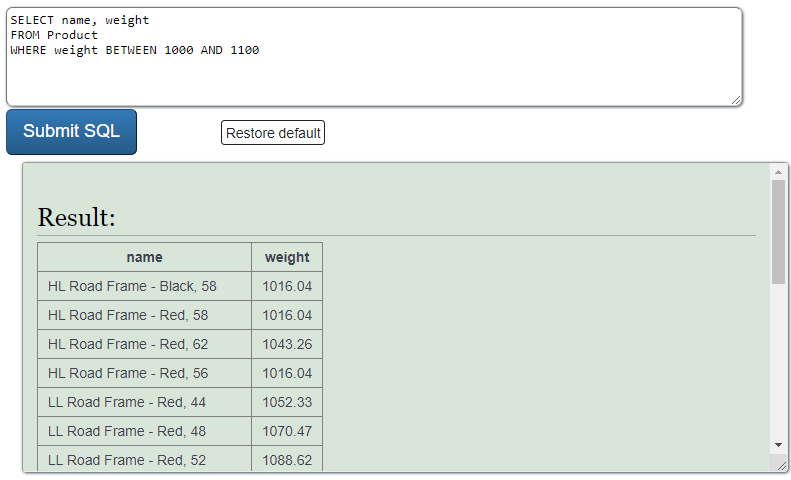


De comparación:

SELECT name, weight

FROM Product

WHERE weight **BETWEEN** 1000 AND 1100

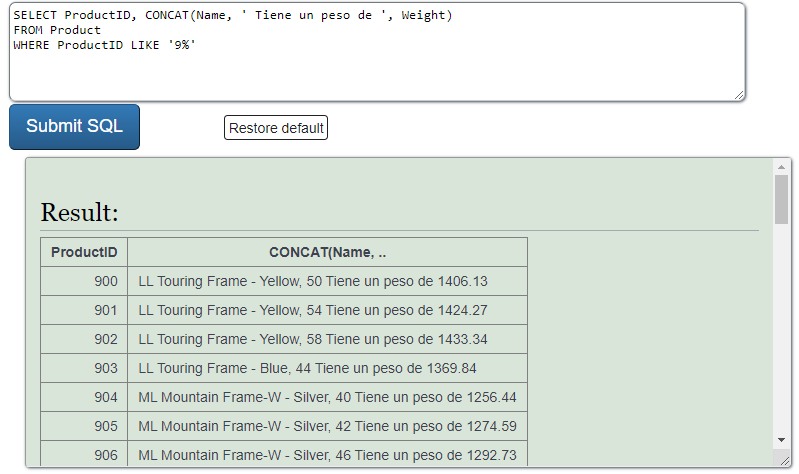


Cadenas:

SELECT ProductID, **CONCAT**(Name, ' Tiene un peso de ', Weight)

FROM Product

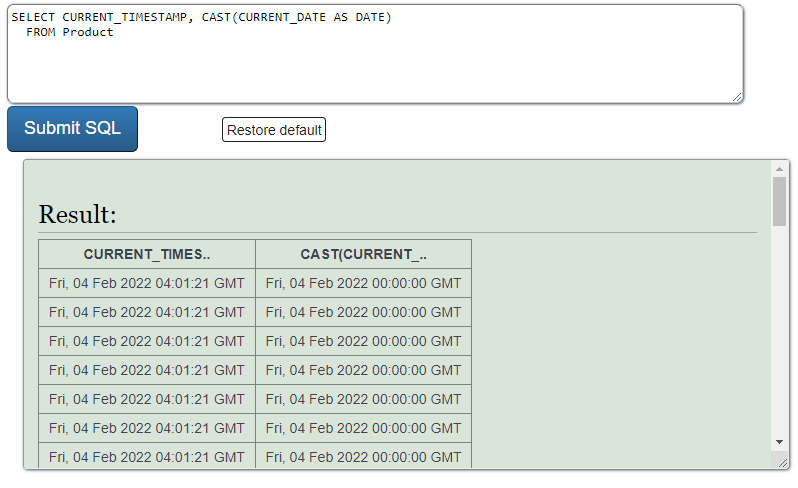
WHERE ProductID **LIKE** '9%'



Tiempo:

SELECT **CURRENT\_TIMESTAMP**, CAST(**CURRENT\_DATE** AS DATE)

FROM Product

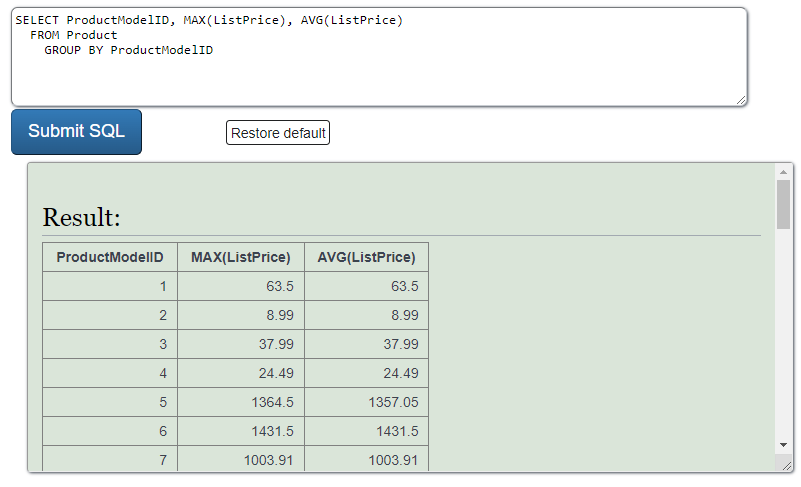


Agrupamiento:

SELECT ProductModelID, **MAX**(ListPrice), **AVG**(ListPrice)

FROM Product

GROUP BY ProductModelID



Condicionales:

SELECT ProductCategoryID, sze

,**CASE** WHEN sze < 50

THEN 'Pequeño'

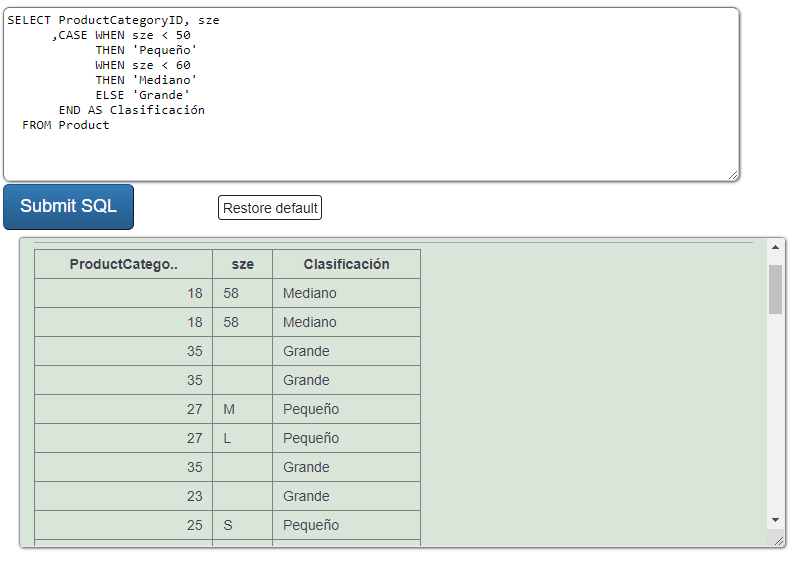
WHEN sze < 60

THEN 'Mediano'

ELSE 'Grande'

END AS Clasificación

FROM Product



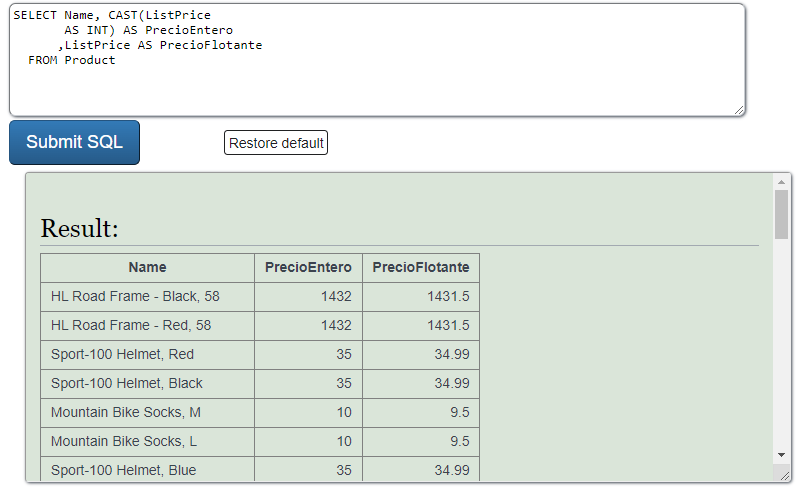
Cambio de tipo:

SELECT Name, **CAST**(ListPrice

AS INT) AS PrecioEntero

,ListPrice AS PrecioFlotante

FROM Product



* 3 consultas anidadas que usen otra consulta:

1) (SELECT …) en FROM

SELECT ProductID, ROUND(Weight) AS Peso

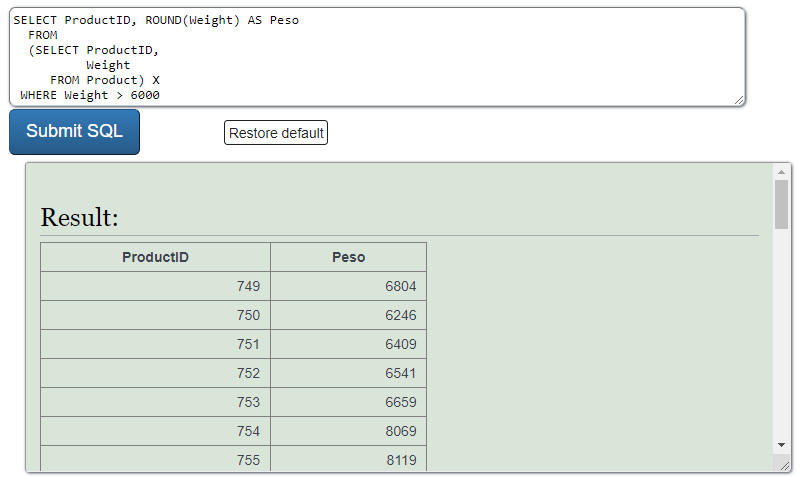
FROM

(SELECT ProductID,

Weight

FROM Product) X

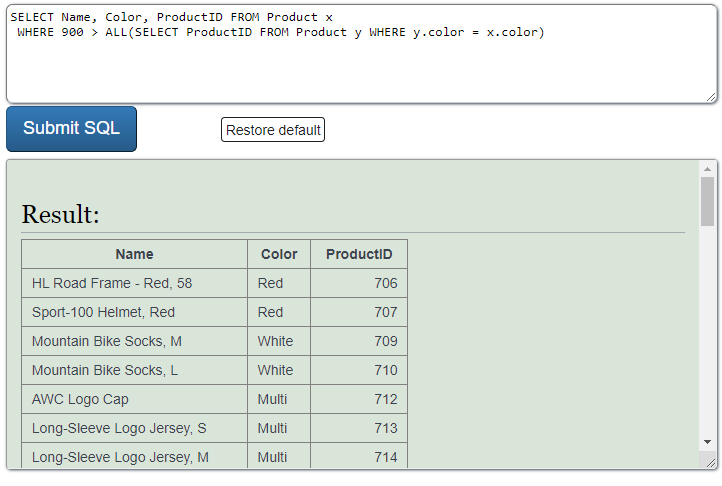
WHERE Weight > 6000



2) SELECT en WHERE

SELECT Name, Color, ProductID FROM Product x

WHERE 900 > ALL(SELECT ProductID FROM Product y WHERE y.color = x.color)

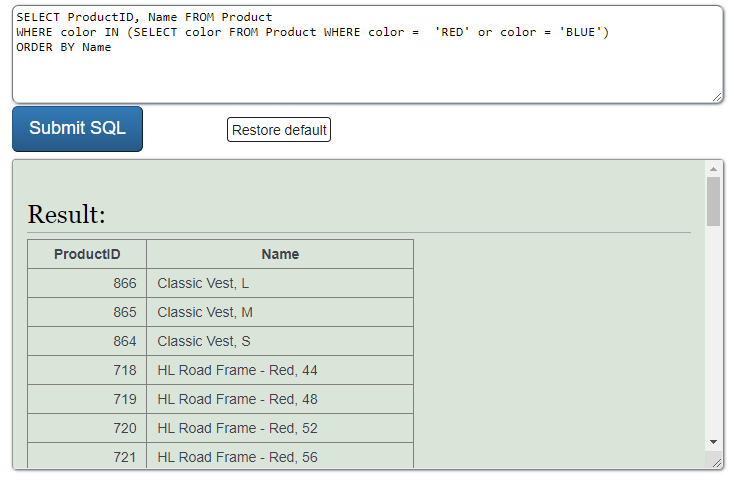


3) SELECT … en SELECT

SELECT ProductID, Name FROM Product

WHERE color IN (SELECT color FROM Product WHERE color = 'RED' or color = 'BLUE')

ORDER BY Name



* 3 consultas con el siguiente esquema:

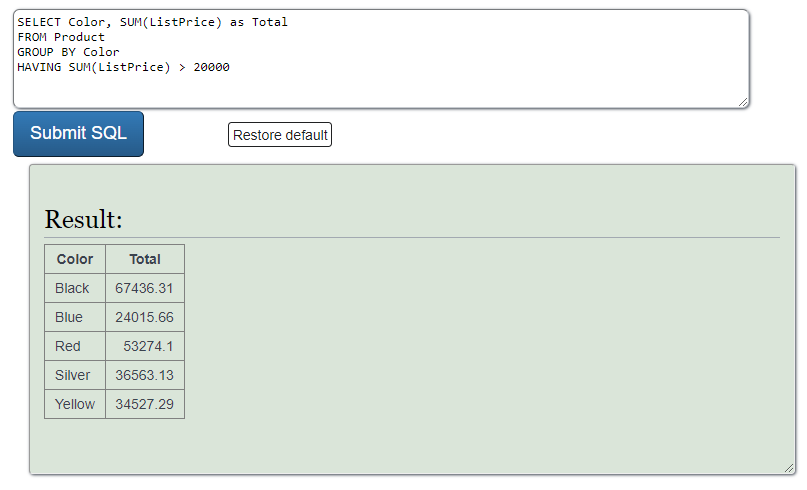
**1)** GROUP BY ... HAVING …

SELECT Color, SUM(ListPrice) as Total

FROM Product

**GROUP BY** Color

**HAVING** SUM(ListPrice) > 20000



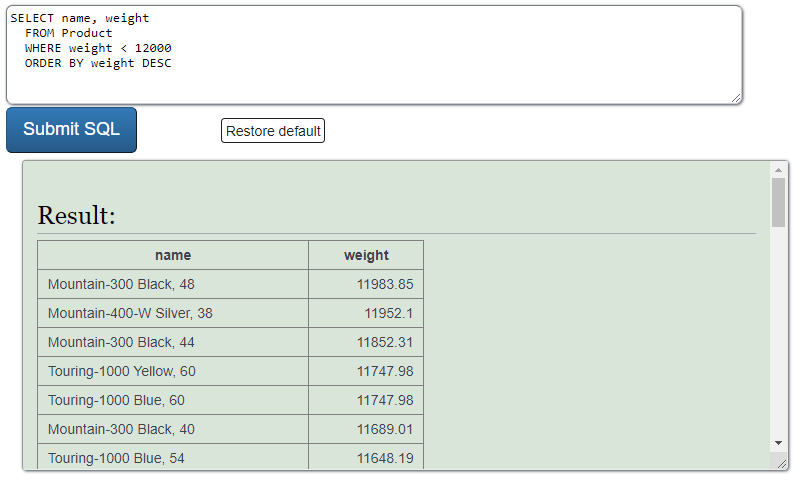
**2)** ORDER BY

SELECT name, weight

FROM Product

WHERE weight < 12000

**ORDER BY** weight DESC



**3)** DISTINCT

SELECT **DISTINCT** color FROM Product

