MODELOS Y BASES DE DATOS

**SQL Básico 2024-1**

**Guía autoestudio 2/6**

# OBJETIVOS

Desarrollar competencias básicas escribir consultas en SQL considerando el valor que representa lo desconocido, operaciones entre conjuntos y juntas explícitas.

# SQL- Detalle

* Dar nuevos nombres a tablas AS
* El valor NULL (DESCONOCIDO)
* Consultas que implican operaciones de conjuntos

UNION, UNION ALL, INTERSECT, EXTRACT, IN

* Consultas que con junta explícita:
  + Junta interna: de equivalencia, natural, cruzada

JOIN, NATURAL JOIN, CROSS JOIN

* + Junta externa: tabla izquierda, tabla derecha, completa

LEFT JOIN, RIGHT JOIN, FULL JOIN

* Operadores
  + Desconocido: ISNULL, COALESCE
  + Lógicos: EXISTS, Comparación ANY, Comparación ALL,
  + Condicionales: CASE

# INVESTIGACIÓN

# NULL

* 1. ¿Qué significa?

SQL usa el término NULL para referirse a la representación de un valor de datos que no existe en la base de datos, es decir, esos valores no representan la cadena vacía o un cero, sino más bien dan a entender que hay la ausencia de un valor o que es desconocido en el campo de datos.

También hay algunas razones que dan a entender que un valor es NULL, por ejemplo:

* El valor no se proporciona durante la entrada de datos.
* El valor aún no se conoce.
  1. ¿Resultado de operarlo con los diferentes tipos de operadores: ¿aritméticos, lógicos y de comparación?

**Aritmético**

* Como sabemos, NULL es la representación de datos ausentes, así:

Suma y Resta (+ y -) = sumar/restar cualquier número con NULL siempre da el mismo número.

Multiplicación (\*) = cualquier número multiplicado por NULL es NULL.

/, DIV, MOD = es imposible porque es como si se estuviera dividiendo por cero.

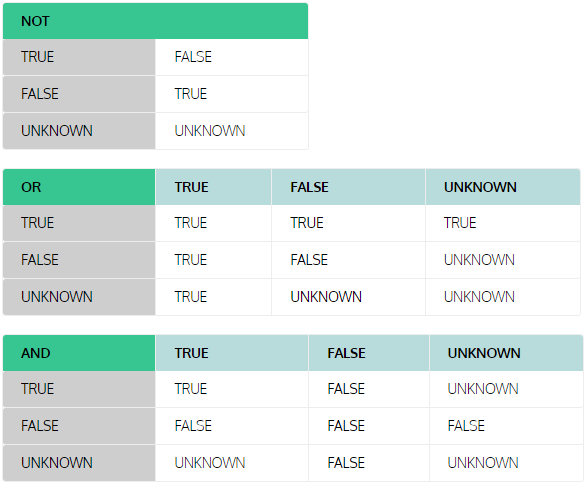
**De comparación**

* Como los valores NULL en general no existen, entonces no es posible usar operadores de comparación (=, !=, <>, >, <, >=, <= , BETWEEN ). Sin embargo, se puede verificar si un valor es NULL con: ISNULL, NOT NULL. El único caso a tener en cuenta es si NULL = NULL es True, y si NULL = 0, entonces es falso.

**Lógicos**

* Los operadores lógicos usan dos valores: True o False (0 o 1). Esos valores en SQL son Verdadero, Falso, Desconocido, por lo tanto, NULL equivale al valor desconocido. Se puede concluir que cualquier operación lógica con NULL da como respuesta un valor desconocido excepto TRUE OR NULL y FALSE AND NULL.

Los usos de estos son:



<https://www.tutorialspoint.com/sql/sql-null-values.htm>

<https://www.ibm.com/docs/es/cognos-analytics/11.0.0?topic=SSEP7J_11.0.0/com.ibm.swg.ba.cognos.ug_cr_rptstd.doc/c_rs_null_values_handling.htm>

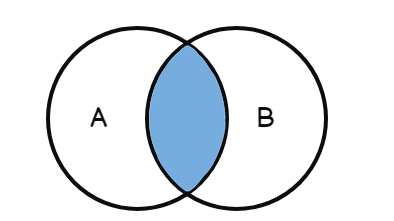
<https://learnsql.es/blog/como-utilizar-los-operadores-de-comparacion-con-null-en-sql/#:~:text=En%20la%20lógica%20ternaria%20de,equivale%20al%20valor%20de%20desconocido.&text=Lo%20que%20hay%20que%20tener,NULL%20y%20FALSE%20AND%20NULL>.

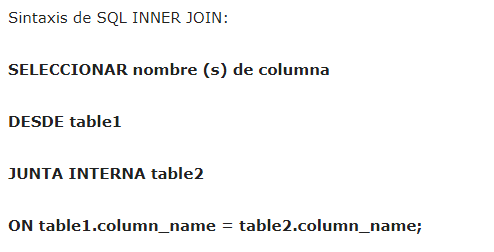
# JUNTA

* 1. ¿Cuáles son las diferencias entre junta interna y externa?

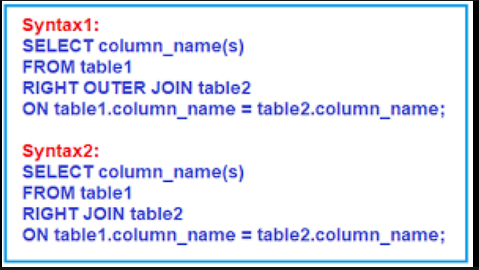
“JOIN es el proceso de tomar datos de varias tablas y colocarlos en una vista generada”. Así, cuando se usa SELECT combina las columnas entre una o más tablas en base de datos relacional y luego va a retornar ese conjunto de datos. También FROM es importante después de usar SELECT para especificar cuál es el origen de los datos.

* **Inner join:** unir dos tablas en una columna común, es decir, tomará los datos comunes que estén en las dos tablas. INTERSECCIÓN. Se basan en los estándares ACID (Atomicidad, consistencia, aislamiento, durabilidad).

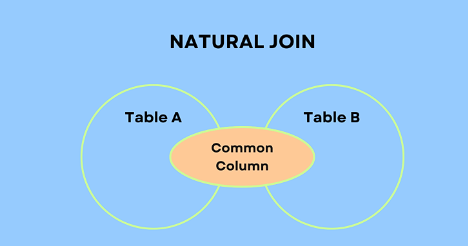
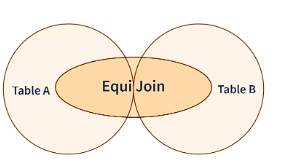


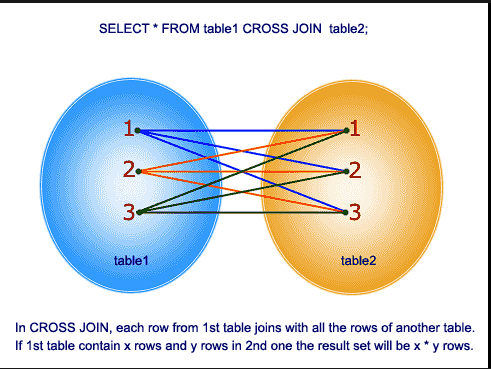


* **Outter join:** “devuelve todas las filas que cumplen la condición de unión y también devuelve algunas o todas las filas de una o ambas tablas para las que ninguna fila satisface la condición de unión”.

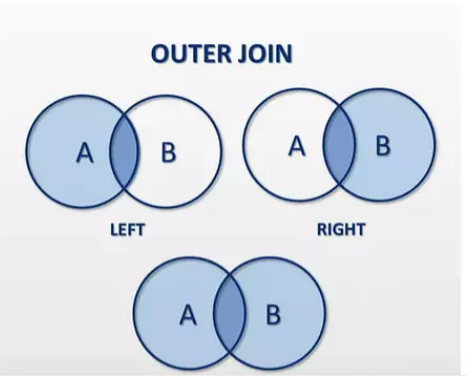


* 1. ¿Qué opciones se tienen para la junta interna?
* **EQUI JOIN:** restringen la condición de la selección a la igualdad de valores de columna.
* **NATURAL JOIN:** dos tablas enlazadas con el mismo nombre.
* **CROSS JOIN**: usada para generar una combinación de parejas ordenadas de cada fila de la primera tabla con cada fila de la segunda tabla. También conocida como CARTESIAN JOIN.





* 1. ¿Qué opciones se tienen para la junta externa?
* LEFT OUTER JOIN: recupera los registros si están presentes en la tabla izquierda.
* RIGHT OUTER JOIN: recupera registros si están presentes en la tabla derecha.
* FULL OUTER JOIN: recupera registros si están presentes en cualquiera de las dos tablas.



<https://www.sqlshack.com/es/introduccion-y-resumen-de-la-clausula-de-sql-join/>

<https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/inner-join/>

<https://www.sqlshack.com/sql-cross-join-with-examples/>

PRACTICA

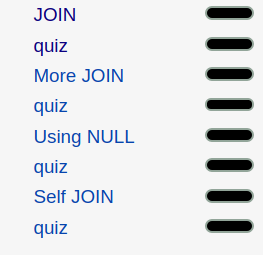
**Usando SQLzoo.net [**[**http://sqlzoo.net/**](http://sqlzoo.net/)**]**

**[En auto02.doc]**

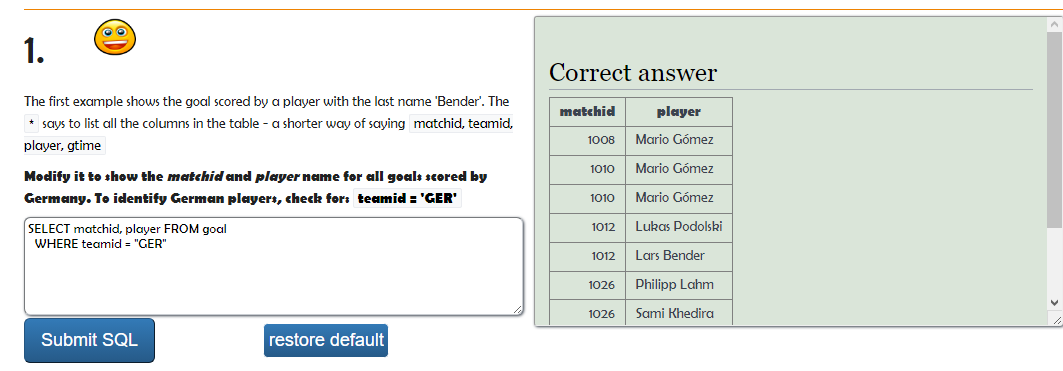
**A.** Realicen los ejercicios propuestos en los siguientes tutoriales. Utilice el motor MySQL.

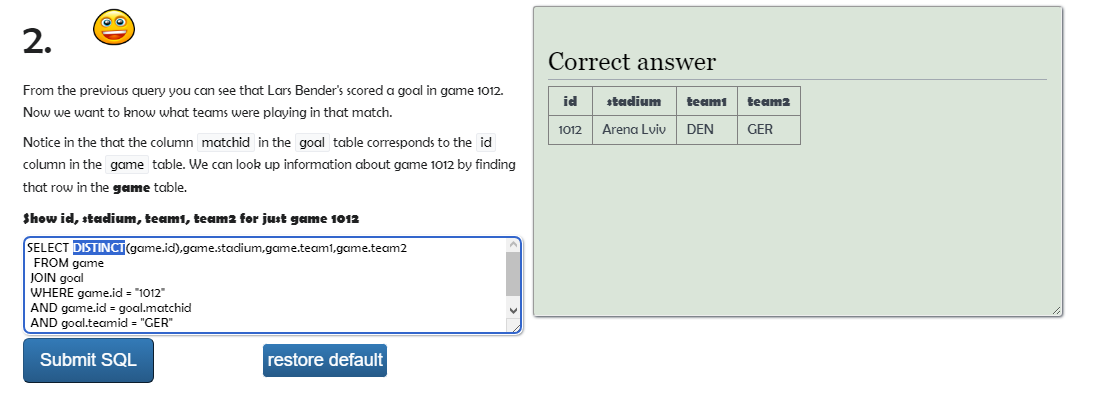
[Para los casos, escriban la sentencia en SQL en auto02.doc. Ejecuten la sentencia SQL en sqlzoo . Si no lograron escribir alguna sentencia indiquen el punto de problema

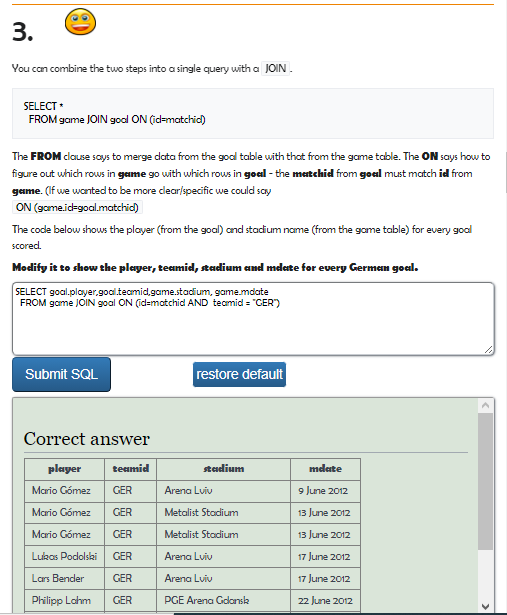
Para los quices, incluyan el puntaje logrado en auto02.doc ]

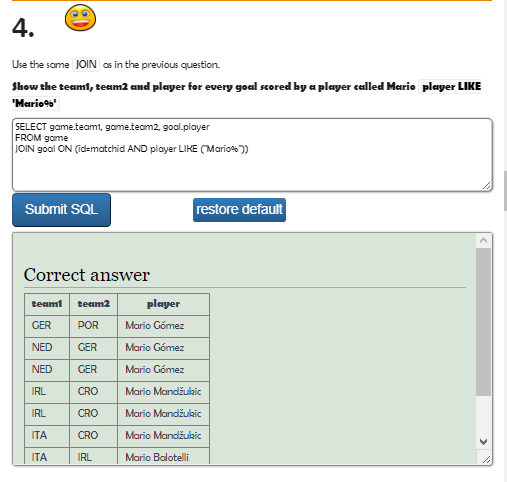


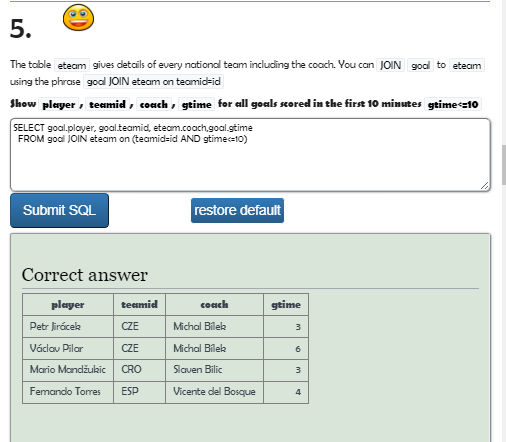


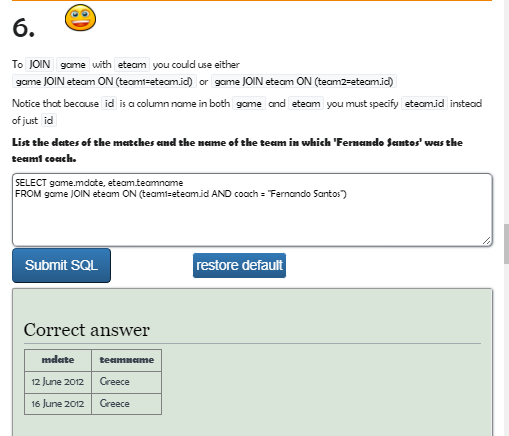


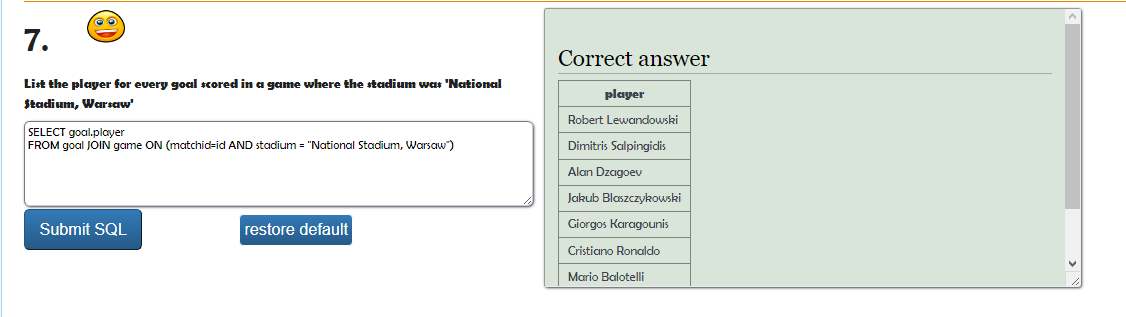


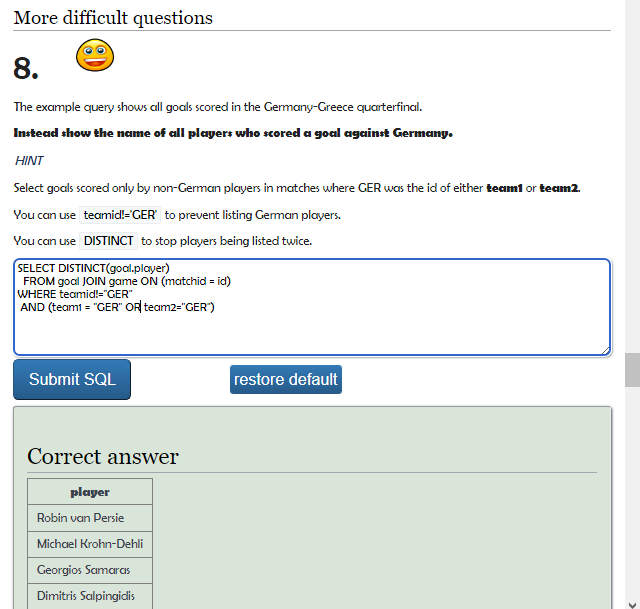


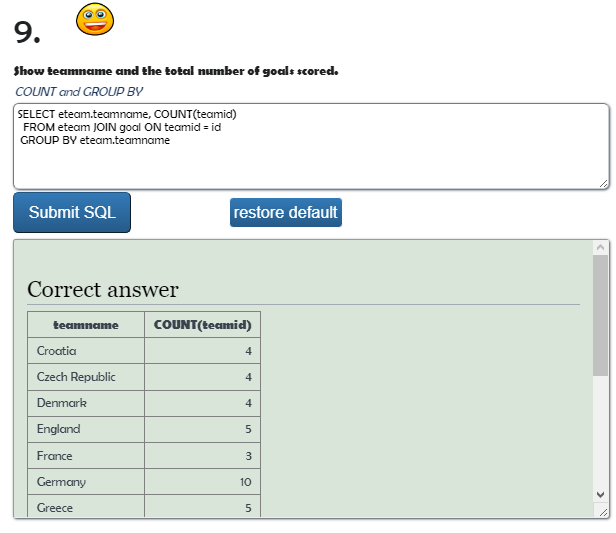


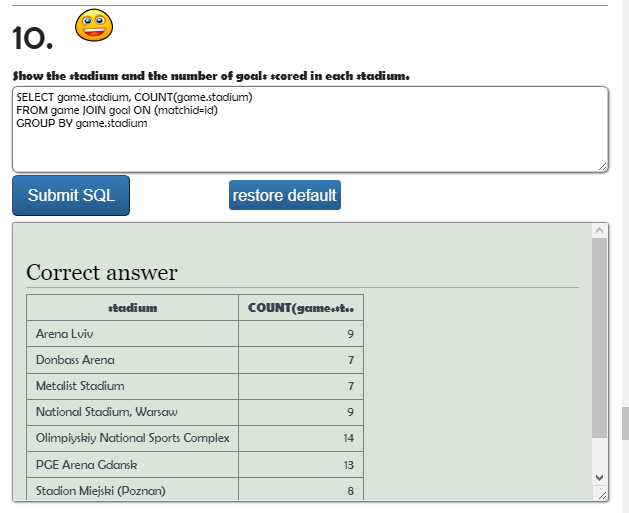


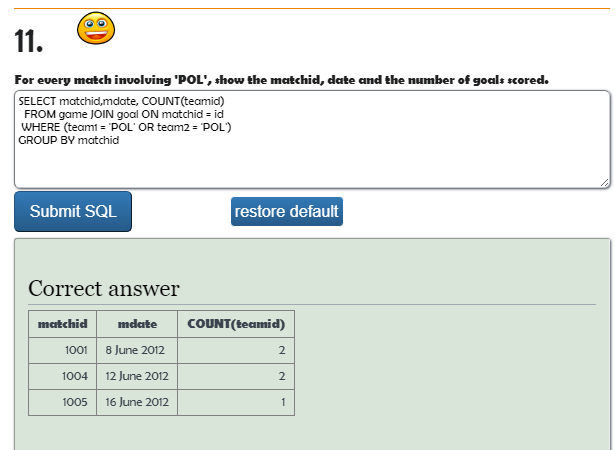


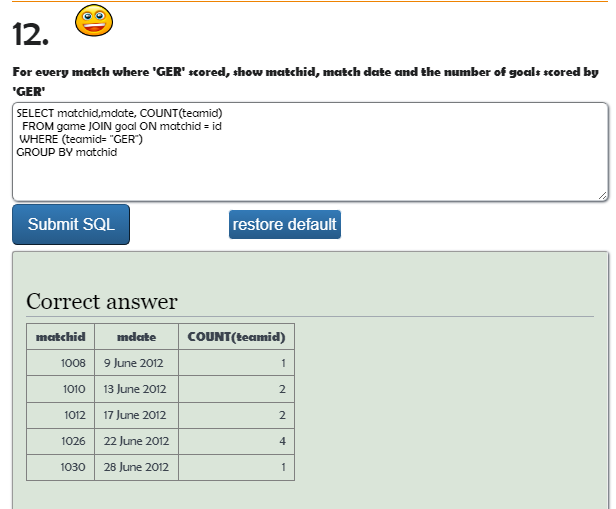


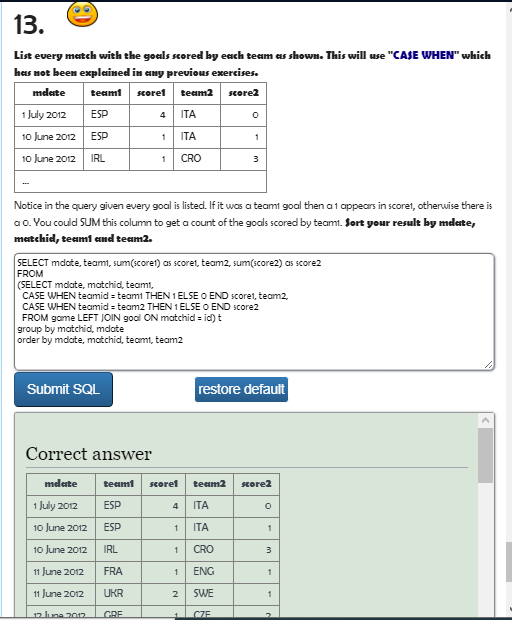




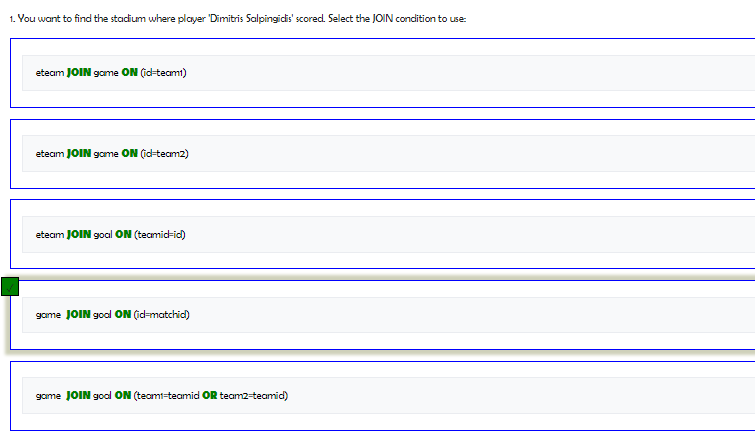


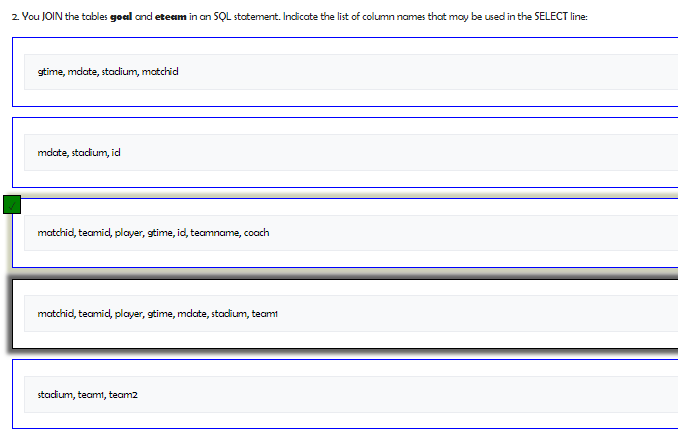


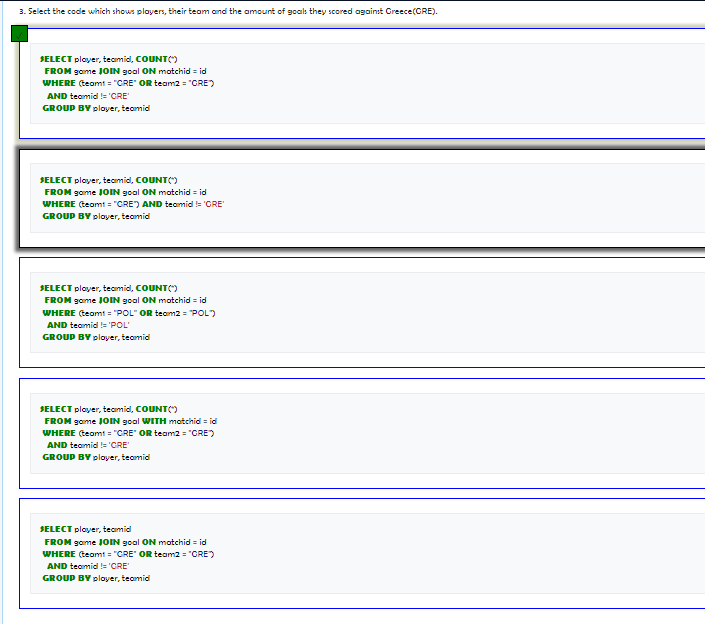


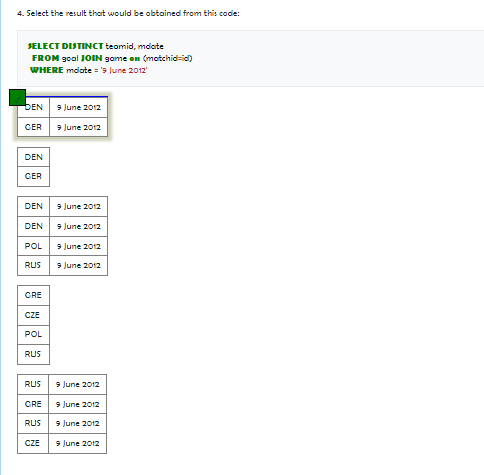


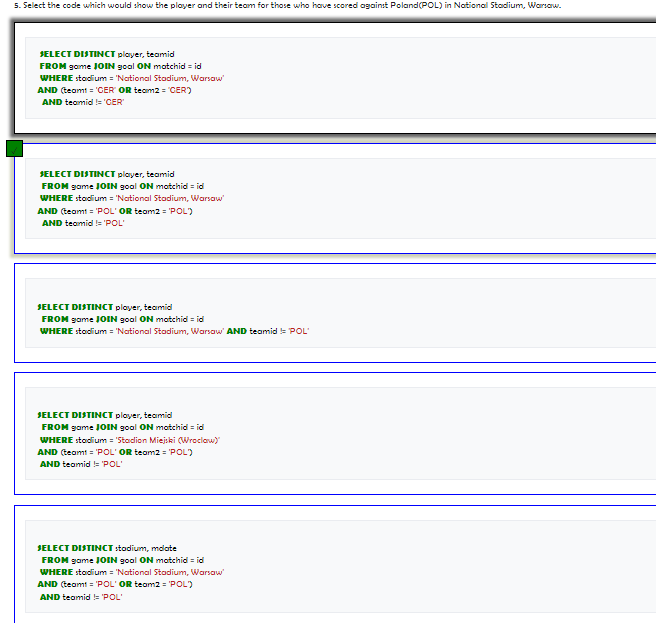




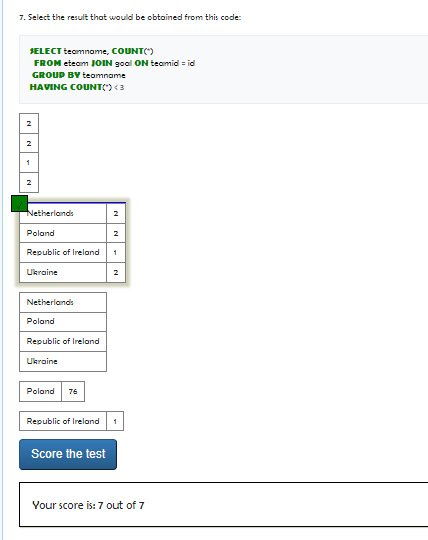




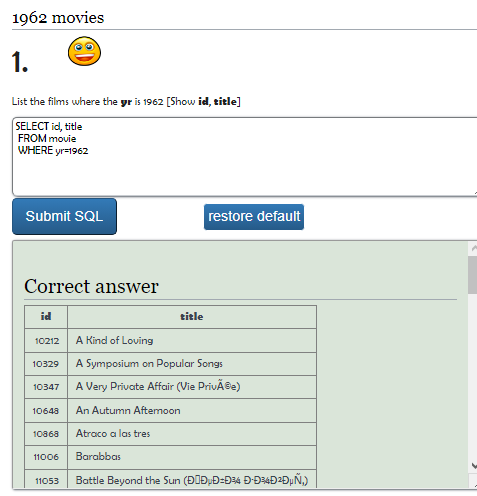


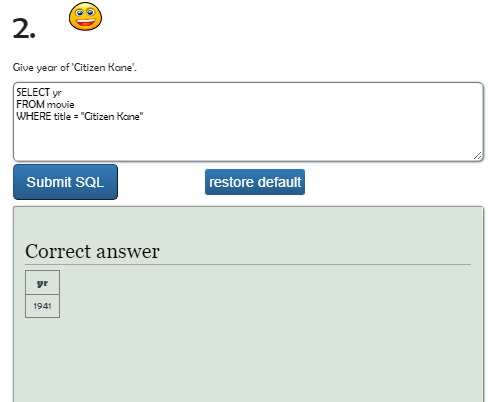


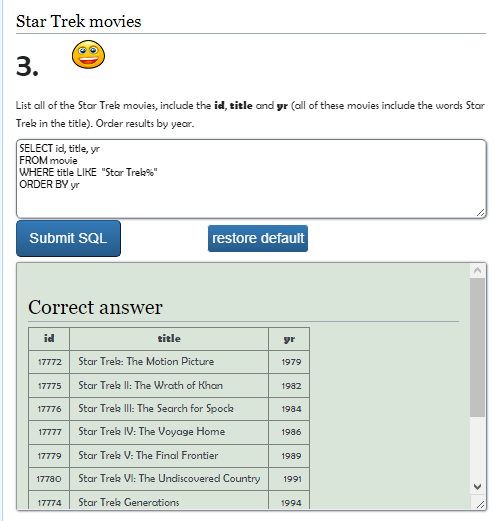


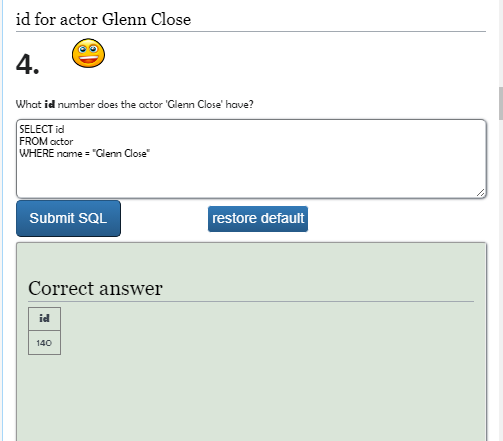


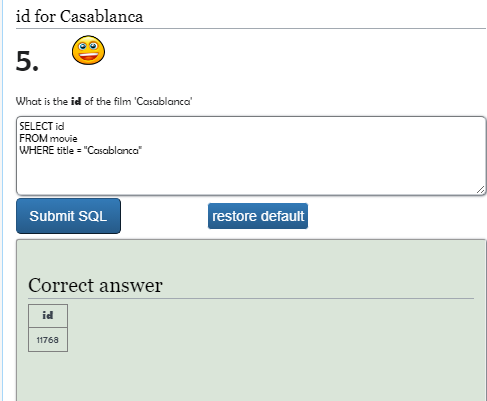


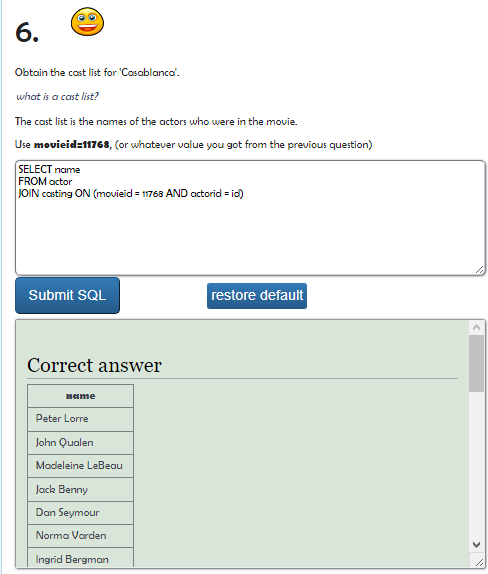


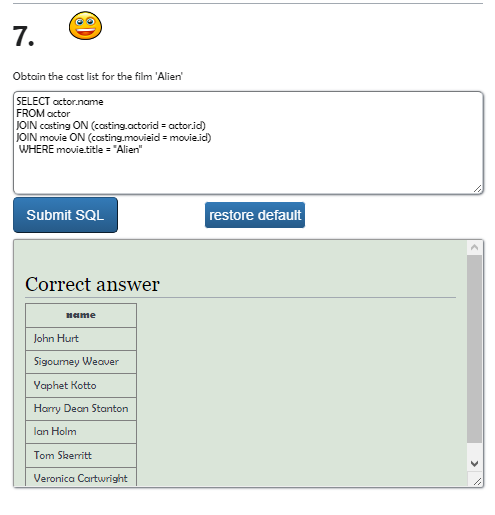


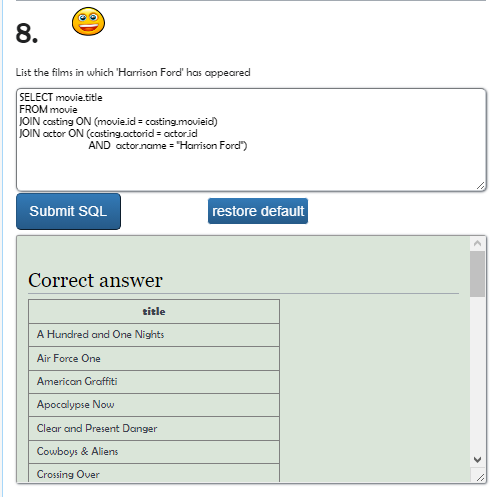


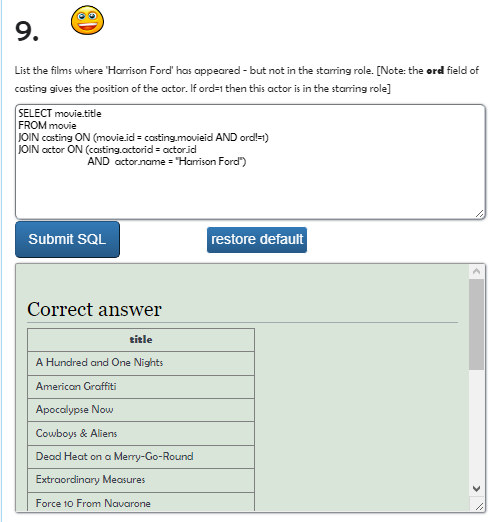


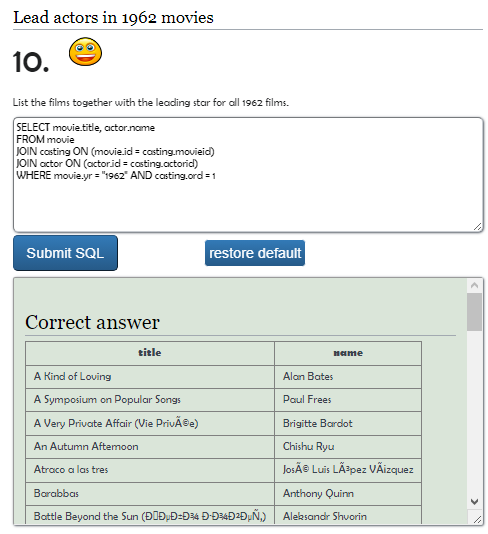


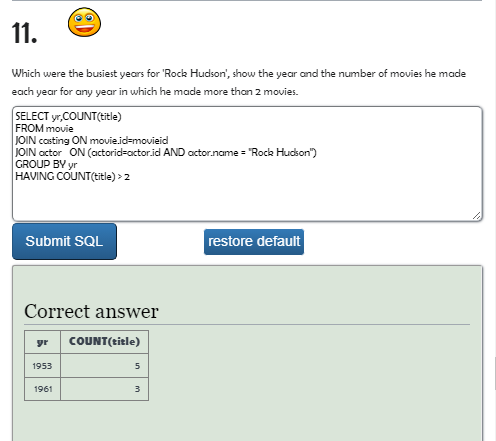


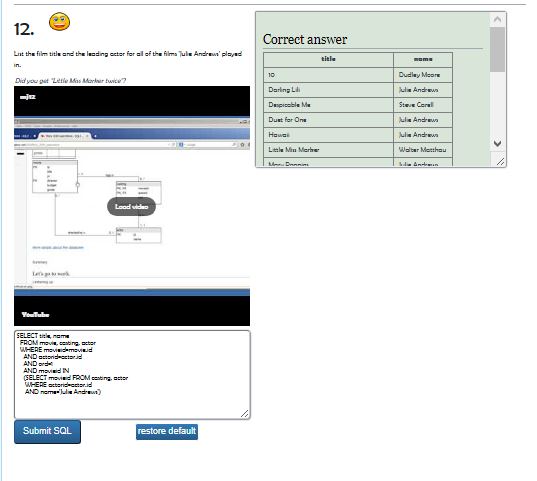


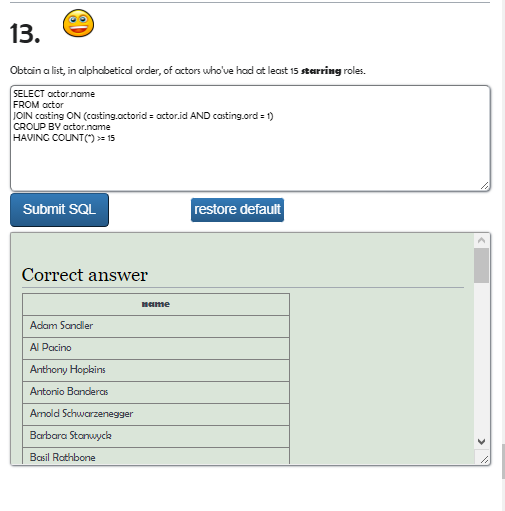


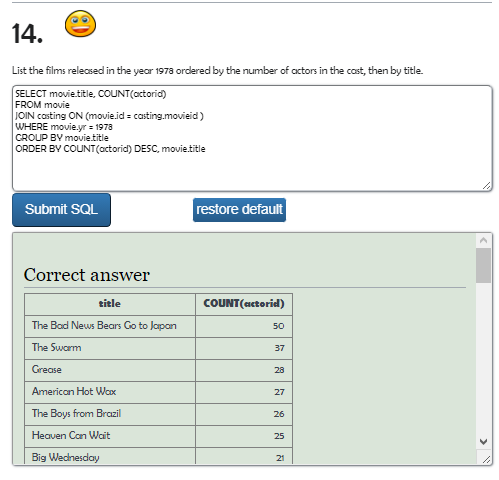


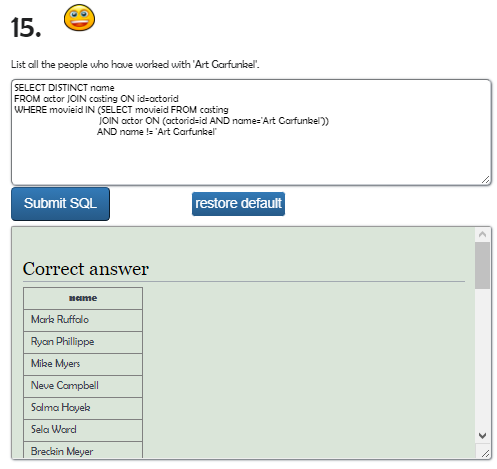






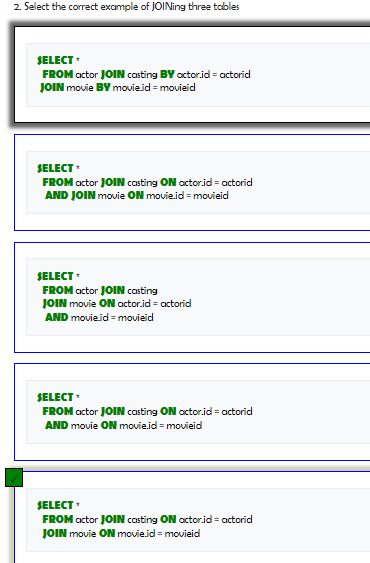




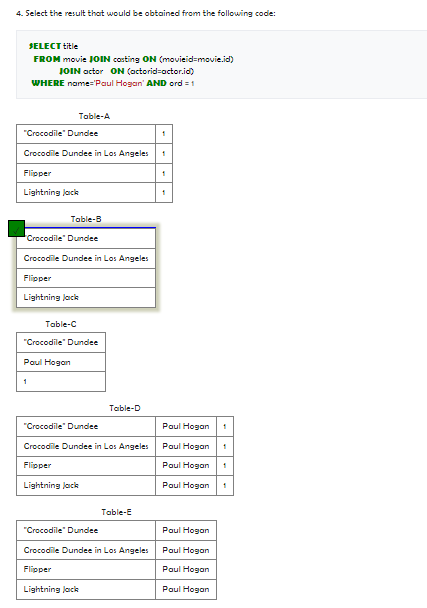


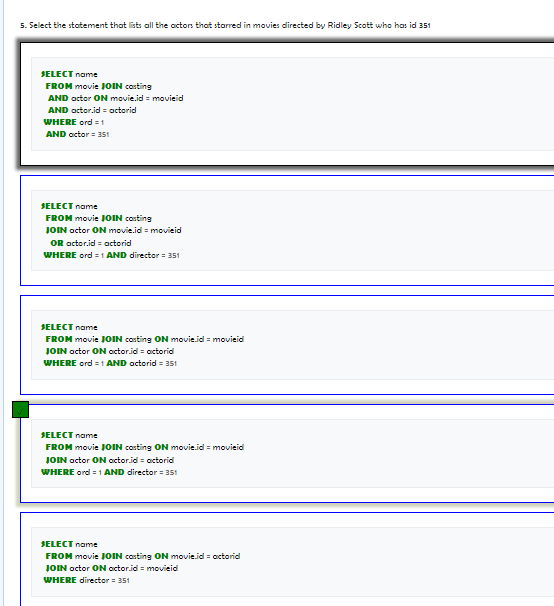


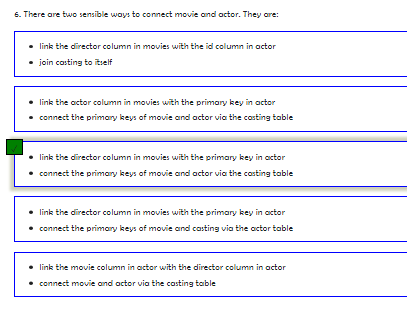


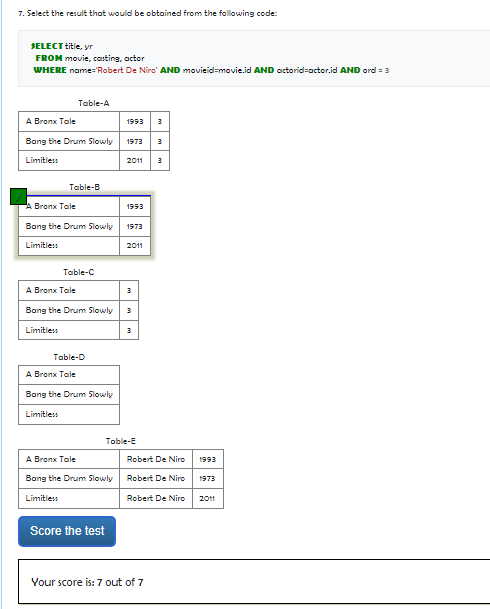




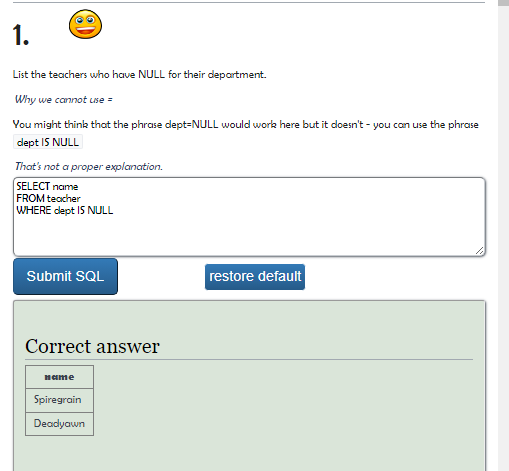


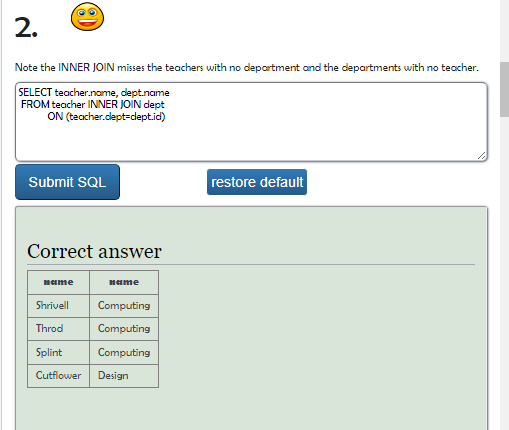


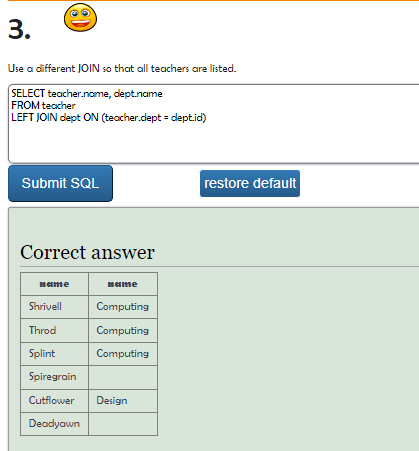


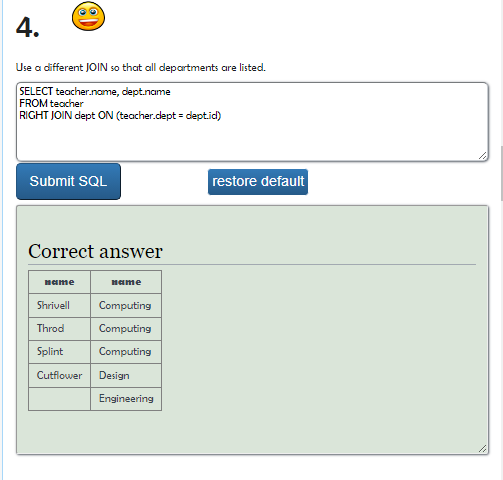


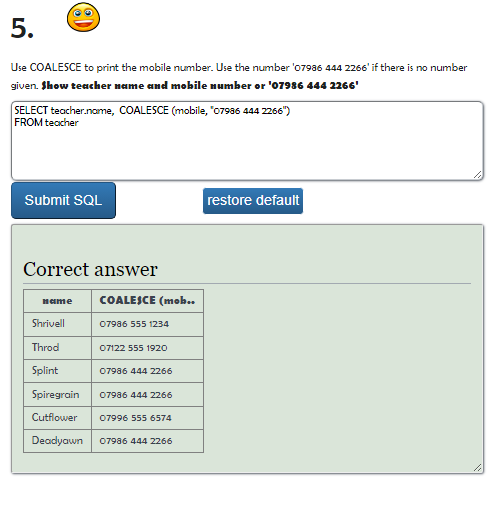


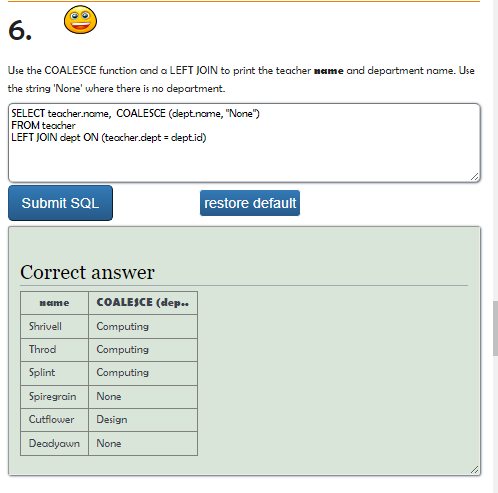


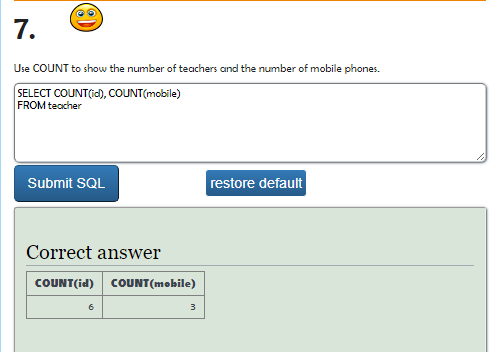


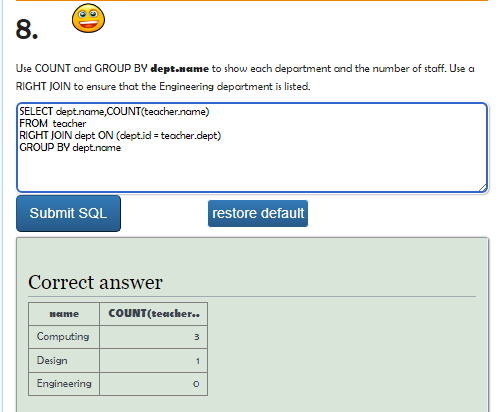


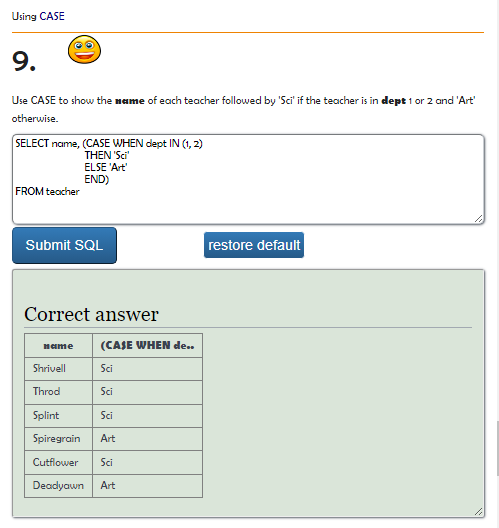


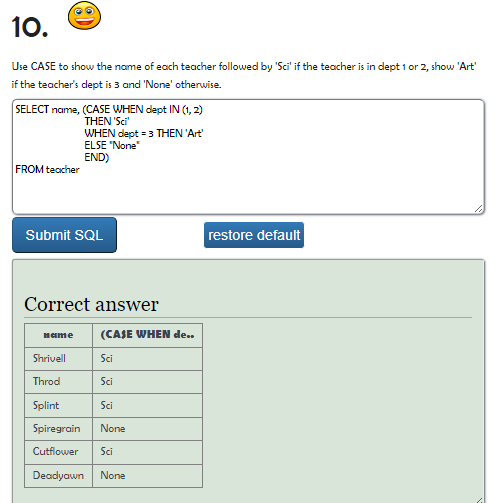






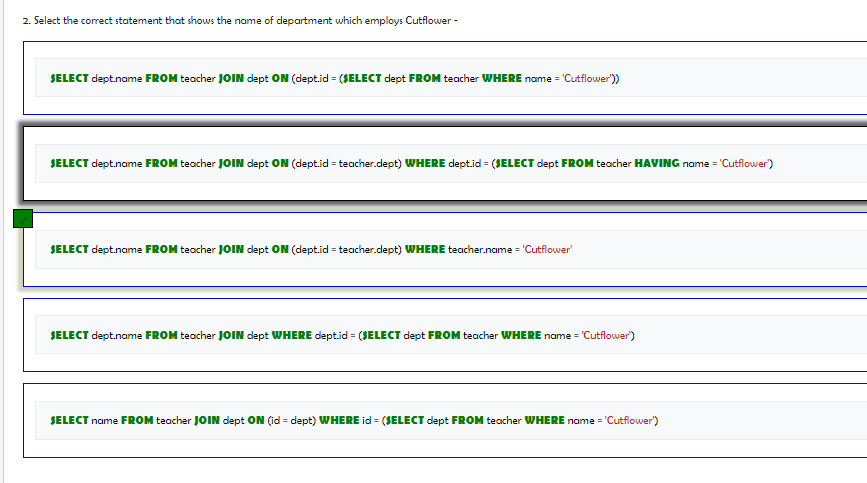


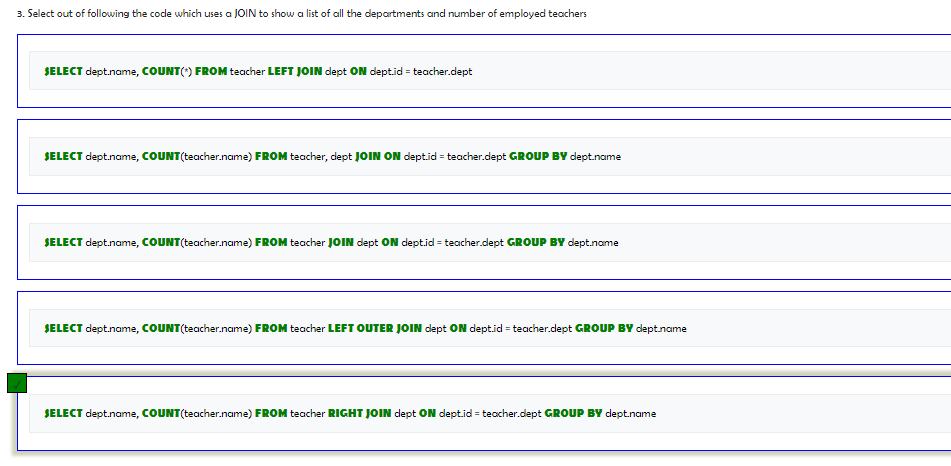


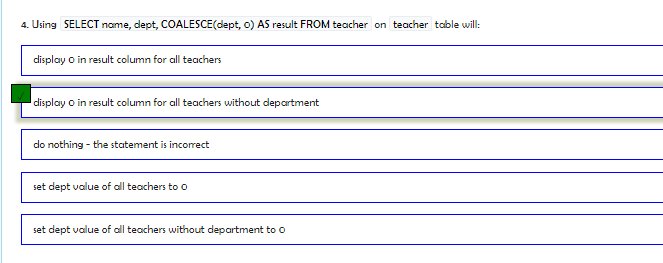


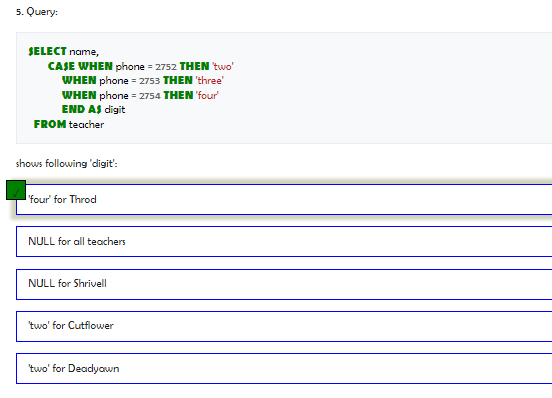




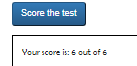




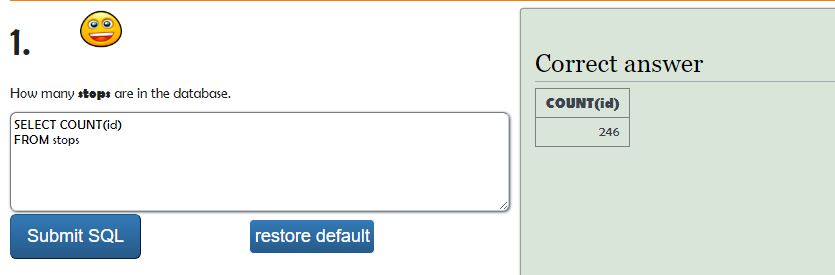




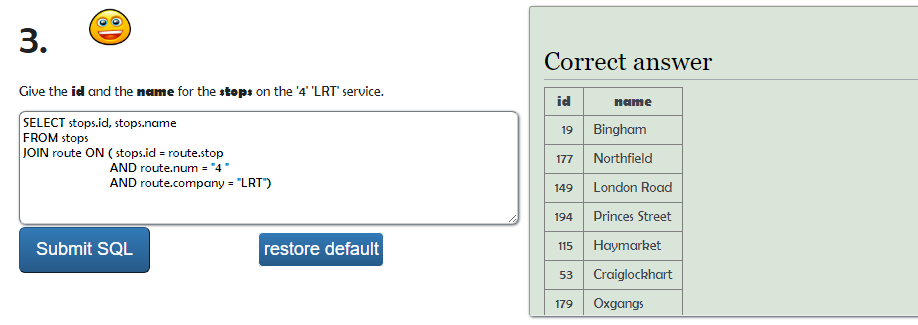


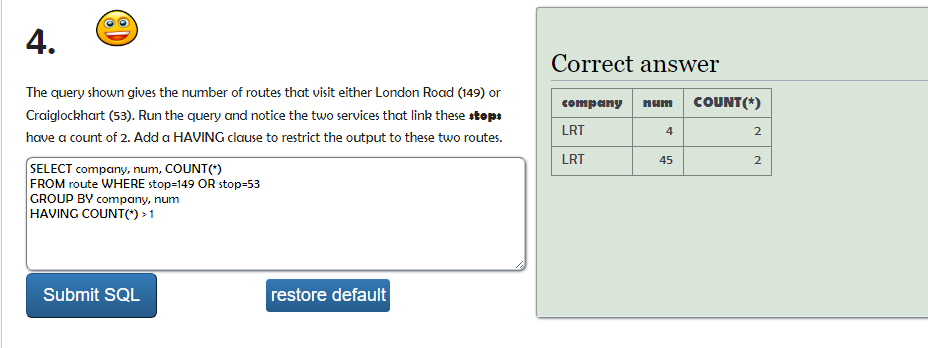


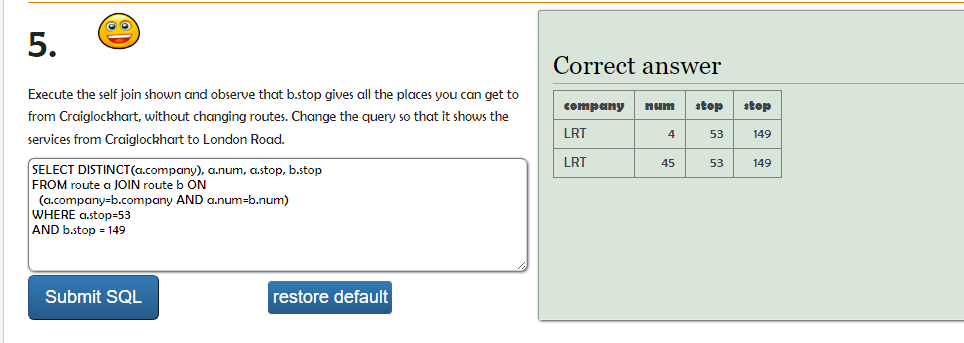


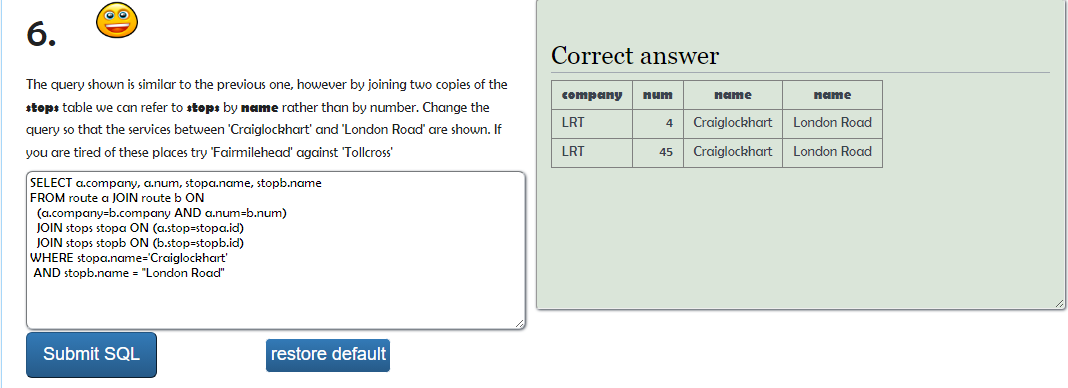


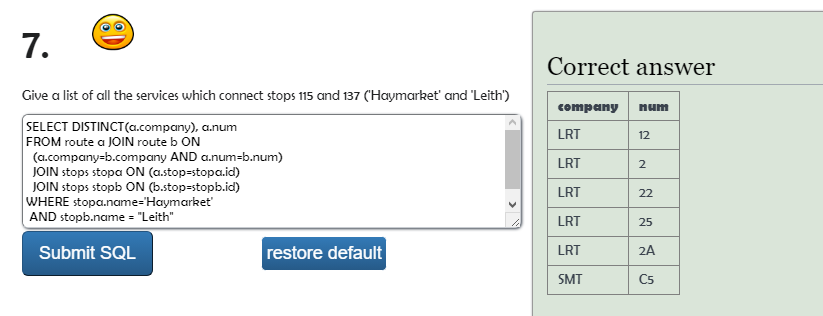




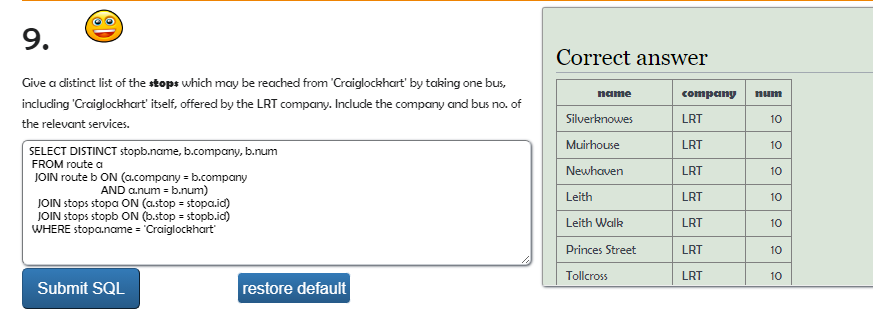


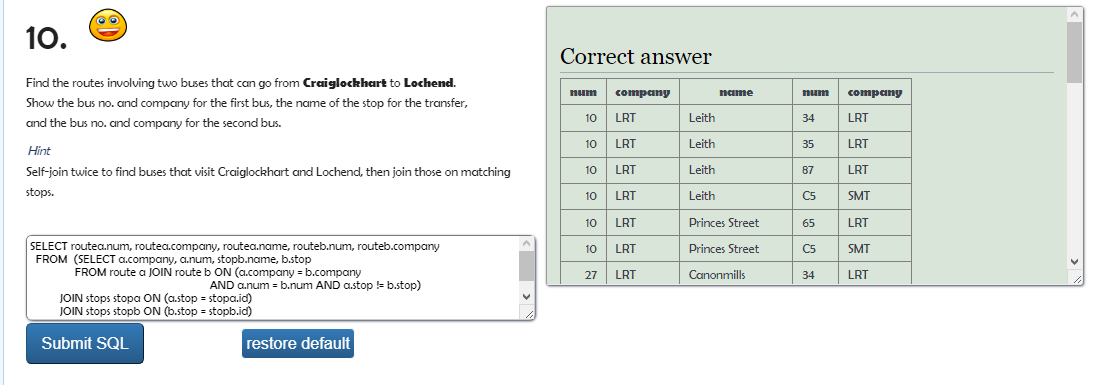


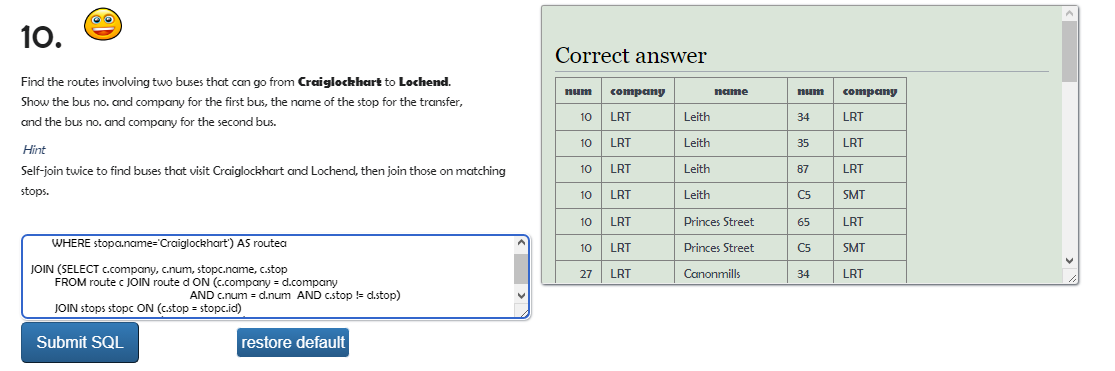


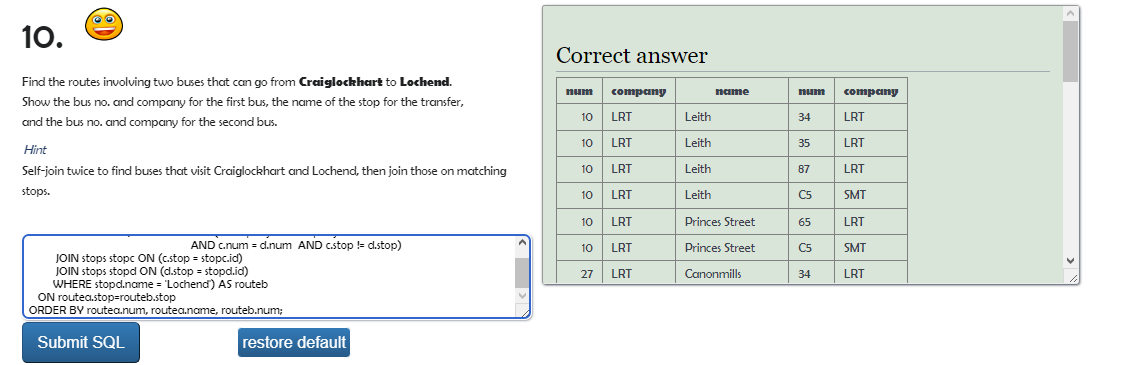










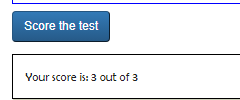












1. Propongan preguntas que cumplan los siguientes requerimientos.

Usen la base de datos [**GuestHouse**](https://sqlzoo.net/wiki/Guest_House)

[Escriban la consulta en lenguaje natural y la sentencia en SQL en auto02.doc. Ejecuten la sentencia SQL en sqlzoo . Si no lograron escribir alguna sentencia indiquen el punto de problema]

* 5 consultas: una para cada operador de conjuntos
* 6 consultas: una para cada operador de junta
* 2 consultas: una para cada operador de desconocido
* 3 consultas: una para cada uno de los tipos de operadores lógicos
* 1 consulta: para el operador CASE

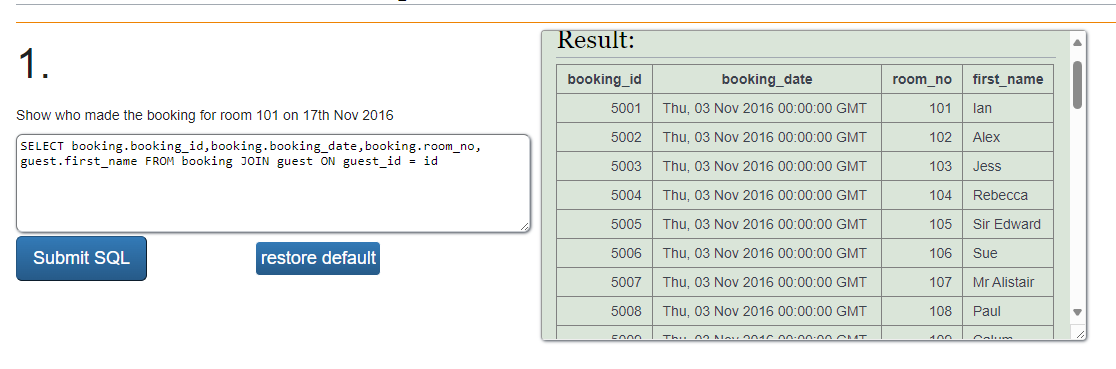
**1. 5 consultas: una para cada operador de conjuntos:**

* **UNIÓN:**

**Lenguaje natural:**

Ver el nombre, el número de cuarto y la fecha de reserva, así como el número de registro de esas reservas.

**SQL:**

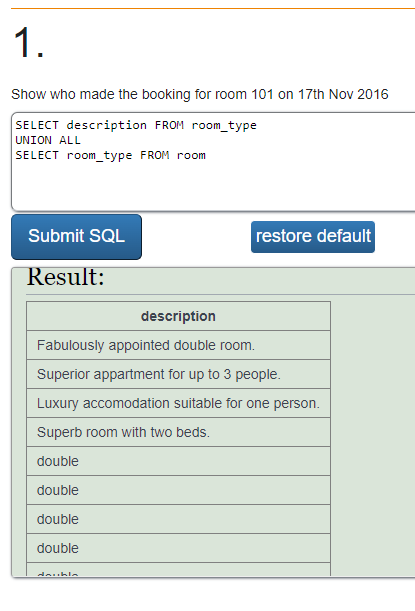


* **UNION ALL:**

**Lenguaje natural:**

Observas los tipos de habitación y descripción de forma secuencial en una sola columna.

**SQL:**

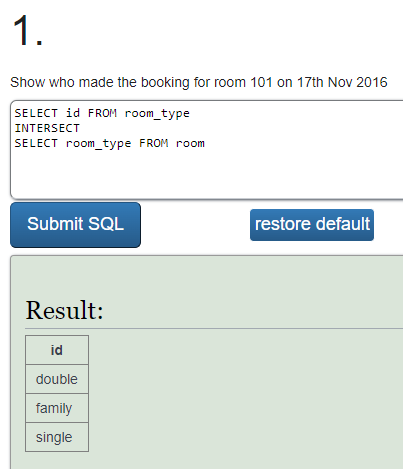


* **INTERSECT:**

**Lenguaje natural:**

Listar los tipos de habitaciones que están en operación actualmente en el hotel.

**SQL:**

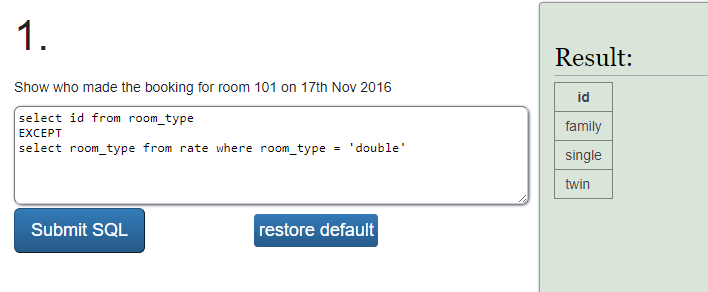
****

* **EXTRACT:**

**Lenguaje natural:**

Se requiere conocer los tipos de habitaciones que hay en el hotel exceptuando las habitaciónes ‘double’ que tienen una tarifa.

**SQL:**

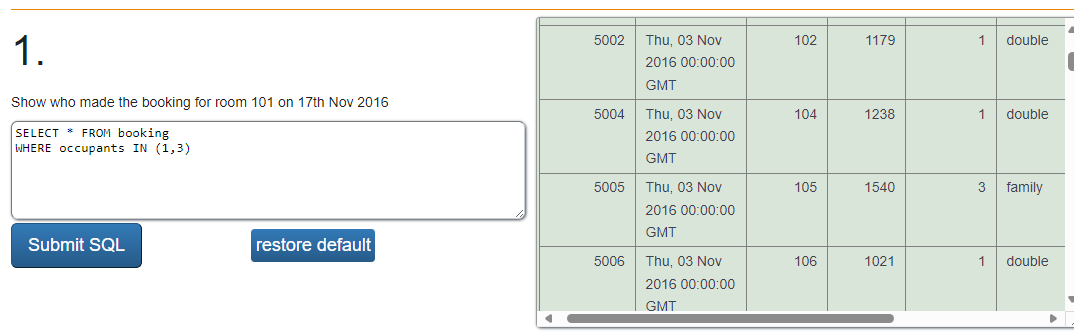
****

* **IN:**

**Lenguaje natural:**

Se quiere conocer todos los datos de todas las reservas en donde hay un 1 y 3 ocupante.

**SQL:**

****

**2. 6 consultas: una para cada operador de junta:**

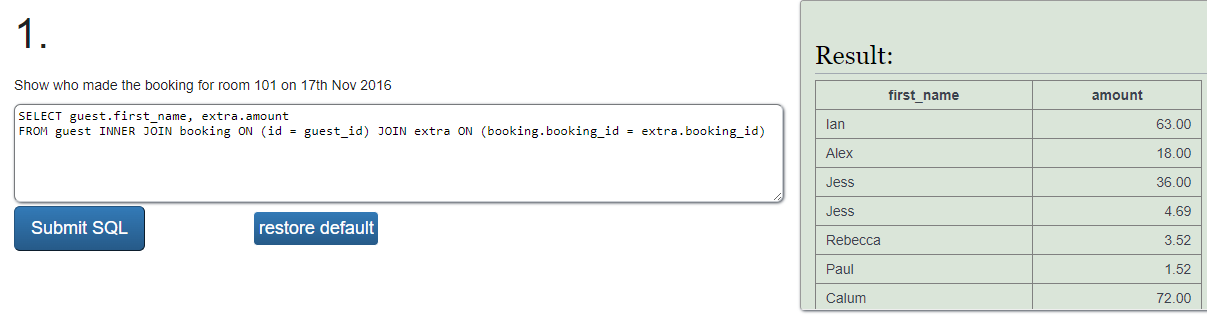
**Junta interna: de equivalencia, natural, cruzada**

* **JOIN:**

**Lenguaje natural:**

Se requiere conocer el nombre del huésped y la tarifa de extras que paga.

**SQL:**

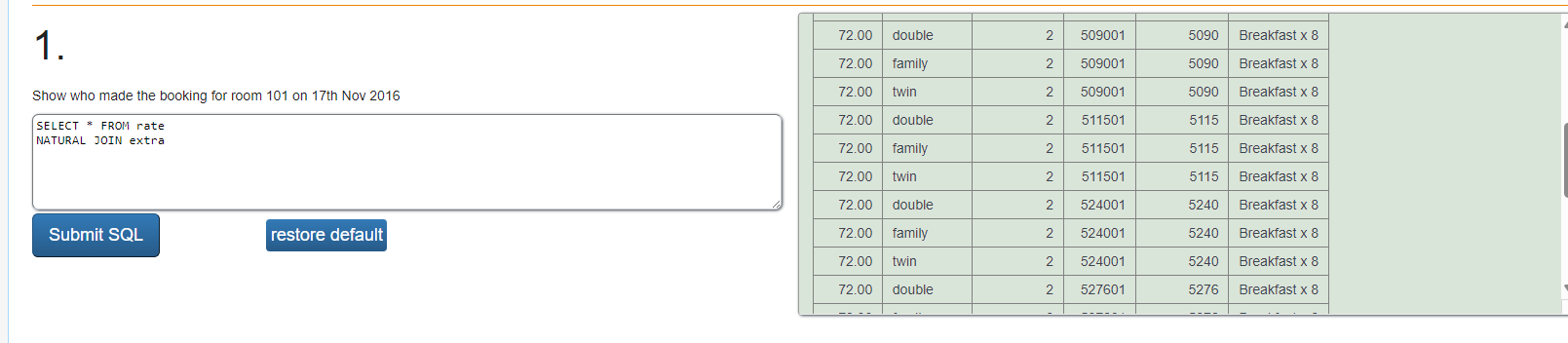
****

* **NATURAL JOIN:**

**Lenguaje natural:**

Se requiere conocer el monto total por cliente tante de extra como de tarifa. No es necesario especificar de cual se trata.

**SQL:**

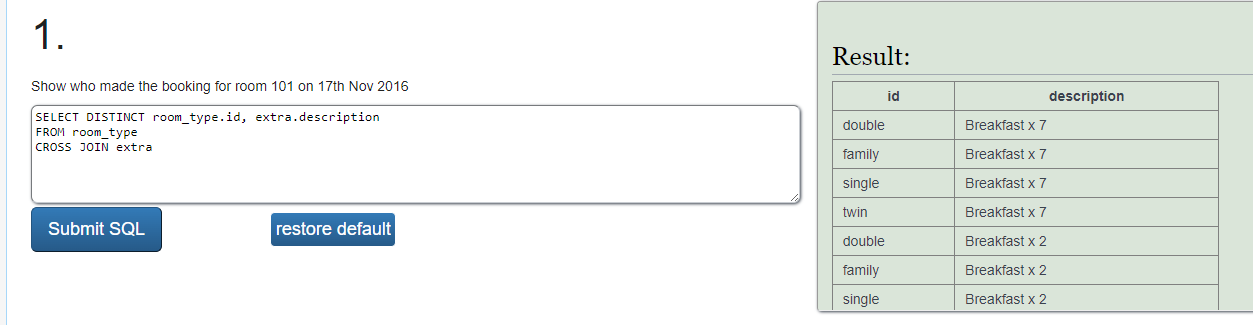
****

* **CROSS JOIN**

**Lenguaje natural:**

Se requiere conocer todas las posibles combinaciones que se pueden generar entre los tipos de habitaciones y los tipos de servicio.

**SQL:**

****

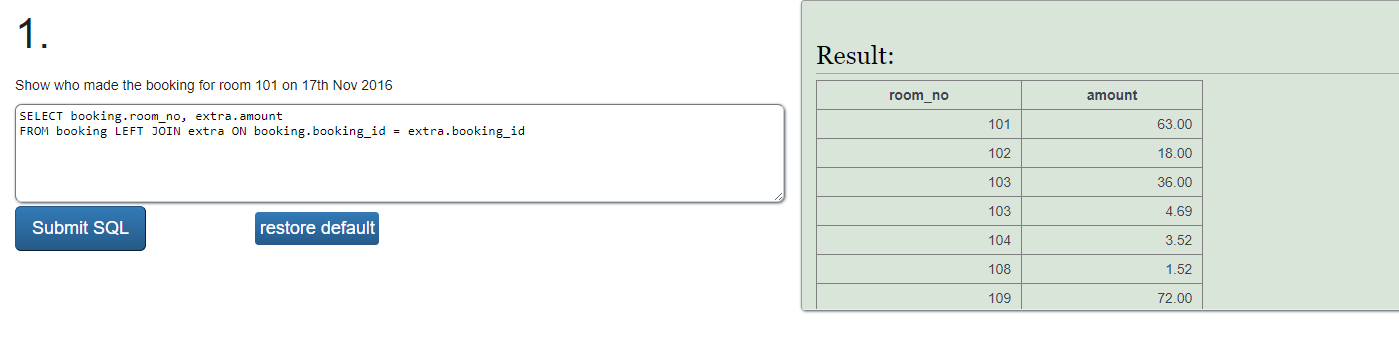
**Junta externa: tabla izquierda, tabla derecha, completa:**

* **LEFT JOIN:**

**Lenguaje natural:**

Se desea conocer el número de habitación y la cantidad de dinero que paga por servicios extra dicha habitación. También se requiere saber si dicha habitación no posee registro de extra.

**SQL:**

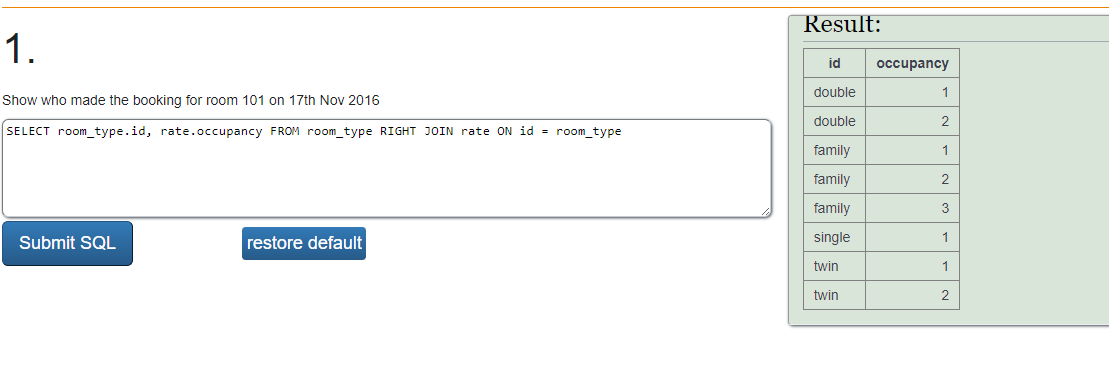
****

* **RIGTH JOIN:**

**Lenguaje natural:**

Se desea saber el tipo de habitación y la cantidad de ocupantes que hay en dicha habitación.

**SQL:**

****

* **FULL JOIN:**

**Lenguaje natural:**

Se desea conocer toda la información hecha por cada reserva más toda la información de la habitación que se tomó por dicha reserva.

**SQL:**

****

**3. 2 consultas: una para cada operador de desconocido.**

* **IFNULL**

**Lenguaje natural:**

Se requiere controlar la excepción de no descripción por servicio dado que algunos pueden ser nulos en descripción y aún así generar costos.

**SQL:**

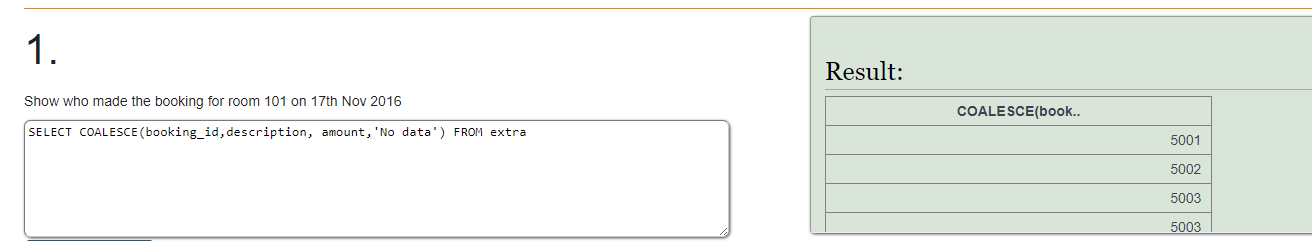
****

* **COALESSCE**

**Lenguaje natural:**

Se desea saber si descripción o tasa de reserva tiene campos no nulos y los que sean nulos asignarles la cadena ‘No data’.

**SQL:**

****

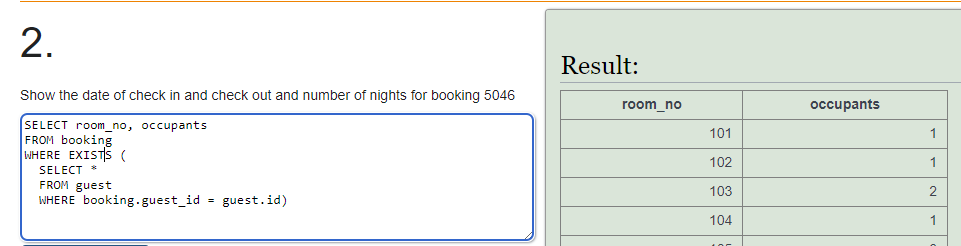
**4. 3 consultas: una para cada uno de los tipos de operadores lógicos.**

* **EXISTS:**

**Lenguaje natural:**

Queremos traer el número de habitación y los ocupantes de las reservas si existe un huésped en la tabla huéspedes.

**SQL:**

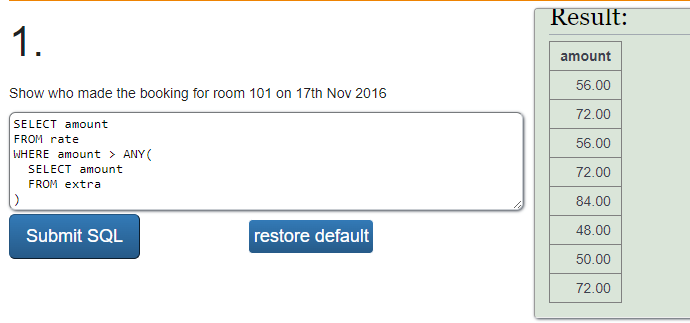
****

* **Comparación ANY:**

**Lenguaje natural:**

Se desea conocer la cantidad de dinero que se está pagando por reserva si el precio de algunas reservas es mayor al precio de los servicios extra.

**SQL:**

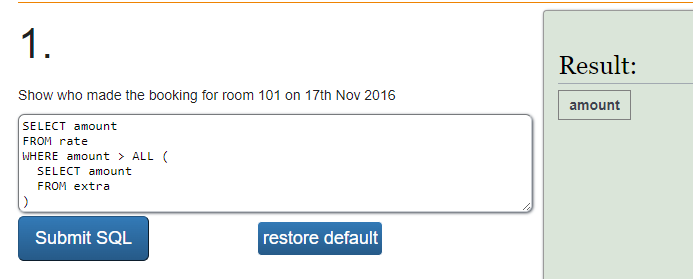
****

* **Comparación ALL:**

**Lenguaje natural:**

Se desea conocer la cantidad de dinero que se está pagando por reserva si el precio solo si la cantidad de dinero por tarifa de reserva de cada reserva es mayor al precio por servicio extra, de lo contrario no se requiere mostrar nada.

**SQL:**

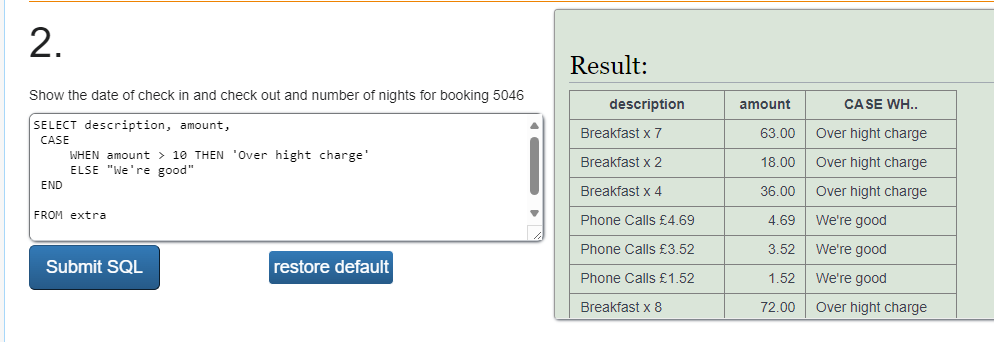
****

**5. consulta: para el operador CASE**

**Lenguaje natural:**

Se quiere saber especificar que, si la cantidad de dinero pagado por servicio extra es mayor a 10 dólares, entonces darle la denominación de sobre costo, de lo contrario informar que se está bien.

**SQL:**

****

No devuelve nada dado que no todos los valores de rate.amount > extra.amount