Pregunta **1**Incorrecta
Se puntúa 0
sobre 1

Marcar
pregunta

inta **2** ecta intúa 1 e 1 arcar

unta

Una clave foránea:

Seleccione una:

- a. Ninguna es correcta.
- $\, \bigcirc \,$ b. No se puede relacionar con una columna cuyo nombre sea distinto.
- O c. No admite valores repetidos.
- od. Exige relacionarse con una clave primaria o clave única.

La respuesta correcta es: Exige relacionarse con una clave primaria o clave única.

inta 3 ecta intúa 1 e 1

unta

 $\ensuremath{\mathrm{c}}$ Qué diferencias hay entre los tipos de datos VARCHAR y NVARCHAR?

Seleccione una:

- o a. Varchar permite almacenar un solo carácter y Nvarchar varios caracteres
- b. Nvarchar permite almacenar el formato Unicode y Varchar no
- oc. Todas son correctas
- O d. Varchar es de longitud fija y Nvarchar de longitud variable

Unicode: Caracteres de diferentes idiomas y escrituras

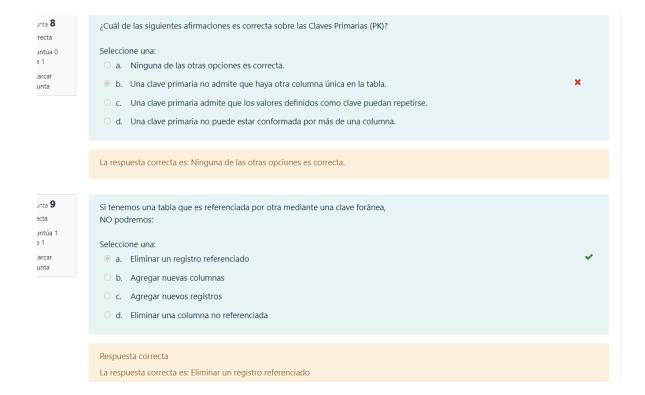
egunta 4 Una clave alternativa es: correcta Seleccione una: puntúa 0 O a. Una clave que verifica el criterio de unicidad. Por lo tanto, no se puede repetir los valores en las columnas definidas como Marcar clave única. egunta Ob. Una clave que se establece con la palabra reservada UNIQUE. oc. Todas las respuestas son correctas. od. Una clave única además de la clave primaria que identifica únivocamente a todo el registro. La respuesta correcta es: Todas las respuestas son correctas. egunta **5** ¿Qué constraint utilizarías para que no se repitan valores en una columna? orrecta Seleccione una: puntúa 1 bre 1 a. CHECK Marcar ob. DEFAULT egunta o c. UNIQUE O d. FOREIGN KEY nta **6** ¿Cuál de las definiciones de Integridad referencial es la correcta? Seleccione una: a como 1 a. No permite ingresar valores en el atributo que referencia, que no se encuentren reflejados en la tabla referenciada donde dicho rcar nta atributo es clave primaria o alternativa. Ob. Permite ingresar valores en el atributo que referencia, que no se encuentren reflejados en la tabla referenciada donde dicho atributo es clave primaria o alternativa. o. No permite ingresar valores en el atributo que referencia, que se encuentren reflejados en la tabla referenciada donde dicho atributo es clave primaria o alternativa. Od. No permite ingresar valores en el atributo que referencia, que no se encuentren reflejados en la tabla referenciada donde dicho atributo es clave primaria. La respuesta correcta es: No permite ingresar valores en el atributo que referencia, que no se encuentren reflejados en la tabla referenciada donde dicho atributo es clave primaria o alternativa. nta **7** Una regla de validación CHECK ta ntúa 1 Seleccione una: 🍥 a. Permite establecer una condición comparando valores de más de una columna siempre y cuando sean de una misma tabla. 💅

O b. Permite establecer una condición comparando valores de más de una columna de una o más tablas de la base de datos.

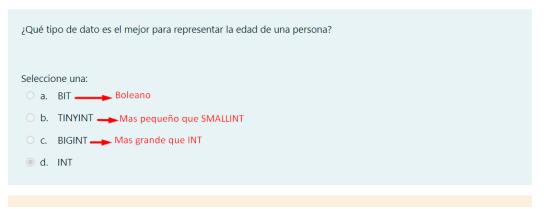
O d. Permite establecer una condición comparando valores de solo una columna de la tabla.

o. Ninguna de las otras opciones son correctas.

rcar



gunta 10 prrecta puntúa 0 re 1 Marcar gunta



En SQL, 'BIGINT' es un tipo de dato numérico utilizado para almacenar números enteros muy grandes. Es una extensión de tamaño para los tipos de datos 'INT' (entero) y 'SMALLINT' (entero pequeño). Un 'BIGINT' generalmente ocupa 8 bytes de almacenamiento, lo que le permite representar números enteros significativamente más grandes que los tipos de datos más pequeños, como 'INT' o 'SMALLINT'.

La siguiente es una breve descripción de algunos tipos de datos numéricos comunes en SQL:

- 'INT': Almacena números enteros dentro de un rango específico, generalmente entre
 -2,147,483,648 y 2,147,483,647 en sistemas de 32 bits. En sistemas de 64 bits, el rango es
 mucho mayor.
- 2. **`BIGINT`**: Tiene un rango aún mayor que **`INT`**, lo que permite almacenar números enteros en el rango de aproximadamente -9,223,372,036,854,775,808 a 9,223,372,036,854,775,807 en sistemas de 64 bits.

El uso de **`BIGINT**` es apropiado cuando necesitas almacenar números enteros muy grandes, como identificadores únicos generados automáticamente, conteos muy grandes o valores que superan el rango de los tipos de datos más pequeños. Sin embargo, es importante considerar el uso adecuado del tipo de dato según los requerimientos de almacenamiento y rendimiento de tu base de datos, ya que el uso excesivo de **`BIGINT**` podría llevar a un mayor consumo de recursos y espacio de almacenamiento.

En SQL, el tipo de dato `TINYINT` es más pequeño que `SMALLINT`. La diferencia principal entre estos dos tipos de datos radica en el rango de valores que pueden almacenar y en la cantidad de espacio de almacenamiento que requieren. Aquí tienes una comparación de ambos:

1. 'TINYINT':

- Rango de valores: -128 a 127 (si se usa con signo) o 0 a 255 (si es sin signo).
- Tamaño de almacenamiento: 1 byte.

2. 'SMALLINT':

- Rango de valores: -32,768 a 32,767 (si se usa con signo) o 0 a 65,535 (si es sin signo).
- Tamaño de almacenamiento: 2 bytes.