

Pregunta 1

Incorrecta

Se puntúa 0  
sobre 1

🚩 Marcar  
pregunta

¿Cuál es el código correcto para generar la siguiente tabla?

+-----+		
	DEMO	
+-----+		
	COL1 ++	
	+	
	COL2 +++++ (PK)	
	+	
	COL3 ++	
+-----+		

Seleccione una:

- ☐ a. 

```
CREATE TABLE DEMO (
    COL1 INT NOT NULL,
    COL2 INT NOT NULL,
    COL3 INT NOT NULL,
    PRIMARY KEY (COL1, COL2, COL3)
)
```

Pregunta 2

Incorrecta

Se puntúa 1  
sobre 1

🚩 Marcar  
pregunta

Una clave foránea:

Seleccione una:

- ☐ a. Ninguna es correcta.
- ☐ b. No se puede relacionar con una columna cuyo nombre sea distinto.
- ☐ c. No admite valores repetidos.
- ☒ d. Exige relacionarse con una clave primaria o clave única.



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

Pregunta 3

Incorrecta

Se puntúa 1  
sobre 1

🚩 Marcar  
pregunta

¿Qué diferencias hay entre los tipos de datos VARCHAR y NVARCHAR?

Seleccione una:

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



Unicode: Caracteres de diferentes idiomas y escrituras

Pregunta 4  
correcta  
Puntuación 0  
de 1  
Marcar  
pregunta

Una clave alternativa es:

Seleccione una:

- ☐ a. Una clave que verifica el criterio de unicidad. Por lo tanto, no se puede repetir los valores en las columnas definidas como clave única.
- ☐ b. Una clave que se establece con la palabra reservada UNIQUE.
- ☐ c. Todas las respuestas son correctas.
- ☒ d. Una clave única además de la clave primaria que identifica unívocamente a todo el registro.



La respuesta correcta es: Todas las respuestas son correctas.

Pregunta 5  
correcta  
Puntuación 1  
de 1  
Marcar  
pregunta

¿Qué constraint utilizarías para que no se repitan valores en una columna?

Seleccione una:

- ☐ a. CHECK
- ☐ b. DEFAULT
- ☒ c. UNIQUE
- ☐ d. FOREIGN KEY



Pregunta 6  
no contestar  
Puntuación 0  
de 1  
Marcar  
pregunta

¿Cuál de las definiciones de Integridad referencial es la correcta?

Seleccione una:

- ☐ a. 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.
- ☐ b. 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.
- ☐ c. 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.
- ☐ d. 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.

Pregunta 7  
no contestar  
Puntuación 1  
de 1  
Marcar  
pregunta

Una regla de validación CHECK

Seleccione una:

- ☒ a. Permite establecer una condición comparando valores de más de una columna siempre y cuando sean de una misma tabla.
- ☐ b. Permite establecer una condición comparando valores de más de una columna de una o más tablas de la base de datos.
- ☐ c. Ninguna de las otras opciones son correctas.
- ☐ d. Permite establecer una condición comparando valores de solo una columna de la tabla.



Pregunta 8

Incorrecta

Puntuación 0

de 1

Marcar

Pregunta

¿Cuál de las siguientes afirmaciones es correcta sobre las Claves Primarias (PK)?

Seleccione una:

- ☐ a. Ninguna de las otras opciones es correcta.
- ☒ b. Una clave primaria no admite que haya otra columna única en la tabla.
- ☐ c. Una clave primaria admite que los valores definidos como clave puedan repetirse.
- ☐ d. Una clave primaria no puede estar conformada por más de una columna.



La respuesta correcta es: Ninguna de las otras opciones es correcta.

Pregunta 9

Correcta

Puntuación 1

de 1

Marcar

Pregunta

Si tenemos una tabla que es referenciada por otra mediante una clave foránea, NO podremos:

Seleccione una:

- ☒ a. Eliminar un registro referenciado
- ☐ b. Agregar nuevas columnas
- ☐ c. Agregar nuevos registros
- ☐ d. Eliminar una columna no referenciada



Respuesta correcta

La respuesta correcta es: Eliminar un registro referenciado

Pregunta 10

Incorrecta

Puntuación 0

de 1

Marcar

Pregunta

¿Qué tipo de dato es el mejor para representar la edad de una persona?

Seleccione una:

- ☐ a. BIT → Boleano
- ☐ b. TINYINT → Mas pequeño que SMALLINT
- ☐ c. BIGINT → Mas grande que INT
- ☒ d. INT

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:

1. `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.