

Punto de control: Resultado

Estimado/a alumno/a:

Rango de NOTA:

0 a 5 - 1
6 a 14 - 2
15 a 17 - 3
18 a 20 - 4
21 a 22 - 5
23 a 24 - 6
25 a 26 - 7
27 a 28 - 8
29 - 9
30 - 10

2P - Variables SQL - CORRECTAS: 0 de 2

1) Variables SQL:

Permiten almacenar un valor y recuperarlo más adelante para emplearlos en otras sentencias. **CORRECTA**

Son específicas de cada consulta y son ejecutadas automáticamente al realizar la conexión.

 Comienzan con "@" (arroba) seguido del nombre (sin espacios), dicho nombre no puede contener cualquier carácter. **INCORRECTA**

Una variable debe ser declarada al final para usarse en el código anterior.

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Variables SQL:

Permiten almacenar un valor y recuperarlo en sentencias anteriores a ser declarada para emplearlos en otras sentencias.

Son específicas de cada consulta y son ejecutadas automáticamente al realizar la conexión.

 Comienzan con "@" (arroba) seguido del nombre (sin espacios), dicho nombre no puede contener cualquier carácter. **INCORRECTA**

Una variable debe ser declarada al final para usarse en el código anterior.

Más de una respuesta de las ANTERIORES es correcta.



1) Variables SQL - Características:

Puede declarar varias variables en una misma sentencia. **CORRECTA**

Existen variables globales en SQL Server.

Una variable declarada existe dentro de todo el entorno de trabajo.

Una variable a la cual no se le ha asignado un valor contiene "0" (Cero).

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Variables SQL - Características:

Solo se puede declarar varias variables en una misma sentencia.

Existen variables globales en SQL Server.

Una variable declarada existe dentro de todo el entorno de trabajo.

Una variable a la cual no se le ha asignado un valor contiene "null". **CORRECTA**

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Procedimientos Almacenados - CORRECTAS: 1 de 2

1) Procedimientos Almacenados:

Es un grupo de sentencias T-SQL compiladas dentro de un plan de una query. **INCORRECTA**

Son un método de encapsular tareas repetitivas que no involucran variables definidas por el motor de SQL para cálculos intermedios.

Son variables de control de flujo de ejecución, para la implementación de bloques condicionales o repetitivos.

Son queries y variables que encapsulan código para su reutilización.

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta. **CORRECTA**

2) Procedimientos Almacenados:



repetitivos.

Son módulos o rutinas que encapsulan código para su reutilización.

- Mas de una respuesta de las ANTERIORES es correcta. **CORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Procedimientos Almacenados - Características - CORRECTAS: 2 de 2

1) Procedimientos Almacenados - Características:

- Aceptan parámetros de entrada. **CORRECTA**

Devuelven mas de un valor de retorno (escalar) que indica el éxito o falla de su ejecución.

No pueden llamar a otros SPs (o sea dentro de un SP no se puede llamar a otros SP).

No pueden devolver valores en la forma de parámetros de salida.

Mas de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Procedimientos Almacenados - Características:

No aceptan parámetros de entrada.

- Devuelven un valor de retorno (escalar) que indica el éxito o falla de su ejecución. **CORRECTA**

No pueden llamar a otros SPs (o sea dentro de un SP no se puede llamar a otros SP).

No pueden devolver valores en la forma de parámetros de salida.

Mas de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Procedimientos Almacenados - Ventajas - CORRECTAS: 1 de 1

1) Procedimientos Almacenados - Ventajas:

Encapsulan la lógica de negocio y crean piezas de código reutilizable por la aplicación.

Todas las aplicaciones pueden usar los mismos procedimientos para asegurar un acceso consistente a los datos.

Evitan la exposición de los detalles de las tablas al usuario, haciendo innecesario el acceso a las tablas en



Todas las respuestas ANTERIORES son correctas. **CORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Procedimientos Almacenados - Parametros - CORRECTAS: 1 de 2

1) Procedimientos Almacenados - Parametros:

Cuando se definen parámetros de entrada y salida, estos siempre van finalizados por el signo @, con nombre del parámetro anteponiéndose, y luego una designación del tipo de dato.

Los parámetros de salida deben incluir la palabra clave OUTPUT para diferenciar de los de entrada.

Los Procedimientos son más flexibles cuando le incluimos parámetros en su definición, de modo tal que pueda crear una lógica de comportamiento más genérica.

Los parámetros de entrada impiden pasar información al SP. Estos valores son ingresados como variables locales al momento de la ejecución dentro del procedimiento almacenado.

Mas de una respuesta de las ANTERIORES es correcta. **CORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Procedimientos Almacenados - Parametros:

Cuando se definen parámetros de entrada y salida, estos siempre van finalizados por el signo @, con nombre del parámetro anteponiéndose, y luego una designación del tipo de dato.

Los parámetros de salida deben incluir la palabra clave PUTOUT para diferenciar de los de entrada.

Los Procedimientos nunca pueden incluir parámetros en su definición, de modo tal que pueda crear una lógica de comportamiento más genérica.

Los parámetros de entrada impiden pasar información al SP. Estos valores son ingresados como variables locales al momento de la ejecución dentro del procedimiento almacenado. **INCORRECTA**

Mas de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta. **CORRECTA**

2P - Tipos de funciones - CORRECTAS: 0 de 2

1) Tipos de funciones - Funciones Escalares:

Devuelven un solo tipo de valor definido en la cláusula "Returns".

Este tipo de funciones es sintácticamente similar a funciones del tipo COUNT(), MAX(). **INCORRECTA**



Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Tipos de funciones - Funciones Tabulares en Linea:

Devuelve una tabla que es el resultado de una única sentencia SELECT. **CORRECTA**

Devuelven un solo tipo de valor definido en la cláusula "Returns".

Es similar a un procedimiento, salvo que puede referenciarse como si se tratara de una tabla.

INCORRECTA

Este tipo de funciones es sintácticamente similar a funciones del tipo COUNT(), MAX().

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Practica - Procedimientos y Funciones - CORRECTAS: 2 de 4

1) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.

```

CREATE PROCEDURE dbo.InsertarOActualizarCelular
    @Marca NVARCHAR(50),
    @Modelo NVARCHAR(50),
    @Memoria NVARCHAR(20),
    @Procesador NVARCHAR(50)
AS
BEGIN
    SET NOCOUNT ON;
    IF EXISTS (SELECT 1 FROM dbo.Inventariocelulares WHERE Modelo = @Modelo)
    BEGIN
        IF EXISTS (SELECT 1 FROM dbo.Inventariocelulares WHERE Modelo = @Modelo AND Marca <> @Marca)
        BEGIN
            THROW 50002, 'Error: El Modelo ingresado ya existe con una Marca diferente.', 1;
        END
        ELSE
        BEGIN
            UPDATE dbo.Inventariocelulares
            SET Memoria = @Memoria,
                Procesador = @Procesador
            WHERE Modelo = @Modelo;
        END
    END
    ELSE
    BEGIN
        INSERT INTO dbo.Inventariocelulares (Marca, Modelo, Memoria, Procesador)
        VALUES (@Marca, @Modelo, @Memoria, @Procesador);
    END
END;

```

Confeccionar un SP que permita insertar un nuevo dispositivo en la tabla dbo.Inventariodispositivos. Si ya existe un registro con el mismo Marca y Modelo, debe actualizar todos los campos (Memoria, Procesador y Marca). Si el Modelo ya está registrado pero la Marca es diferente, debe actualizar solo los valores de Memoria y Procesador sin generar ningún error.

Confeccionar un SP que permita insertar un nuevo celular en la tabla dbo.Inventariocelulares. Si ya existe un registro con el mismo Modelo pero con una Marca diferente, se debe permitir la inserción de un nuevo registro sin



procesador, y si el valor ingresado es una Marca y Modelo diferente a los ya cargados se debe cargar el nuevo registro, y si los datos son duplicados se generar el error. **CORRECTA**

Confeccionar un SP que permita insertar un nuevo celular en la tabla dbo.Inventariocelulares. Si ya existe un registro con el mismo Modelo, se debe actualizar solo el campo Marca y dejar intactos los valores de Memoria y Procesador. Si se intenta insertar un nuevo registro con un Modelo y Marca duplicados, el procedimiento debe permitir la inserción sin lanzar ningn error.

Ninguno de los enunciados ANTERIORES es correcto. Código Incorrecto

2) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.

```
CREATE TABLE dbo.NumeroPar (
    nro SMALLINT
);
CREATE PROCEDURE InsertarNumerosPares
AS
BEGIN
    DECLARE @i SMALLINT;
    SET @i = 4;
    WHILE @i <= 22
    BEGIN
        IF @i NOT IN (8, 12, 18)
        BEGIN
            INSERT INTO dbo.NumeroPar (nro) VALUES (@i);
        END
        SET @i = @i + 2;
    END
END;
```

Creacion de Tabla dbo.NumeroPar y Store Procedure que permita insertar números pares del 4 al 22 en una tabla con el nombre dbo.NumeroPar (nro smallint). Excepto los números 8, 12 y 18. La tabla debe ser creada en el procedimiento. **INCORRECTA**

Creacion de Tabla dbo.NumeroPar y Store Procedure que permita insertar números pares del 4 al 21 en una tabla con el nombre dbo.NumeroPar (nro smallint). Excepto los números 8, 12 y 18. La tabla debe ser creada fuera del procedimiento.

Creacion de Tabla dbo.NumeroPar y Store Procedure que permita insertar números pares del 4 al 21 en una tabla con el nombre dbo.NumerosPar (nro smallint). Excepto los números 8, 12 y 18. La tabla debe ser creada fuera del procedimiento.

Creacion de Tabla dbo.NumeroPar y Store Procedure que permita insertar números pares del 4 al 21 en una tabla con el nombre dbo.NumeroPar (nro smallint). Excepto los números 8, 12 y 18. La tabla debe ser creada fuera del procedimiento.

Ninguno de los enunciados ANTERIORES es correcto. Código Incorrecto **CORRECTA**

3) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.



```

  BEGIN
    DECLARE @i SMALLINT;
    SET @i = 4;
    WHILE @i < 22
    BEGIN
      IF @i NOT IN (8, 12, 18)
      BEGIN
        INSERT INTO dbo.NumeroPar (nro) VALUES (@i);
      END
      SET @i = @i + 2;
    END
  END;

```

Creacion de Tabla dbo.NumeroPar y Store Procedure que permita insertar números pares del 4 al 22 en una tabla con el nombre dbo.NumeroPar (nro smallint). Excepto los números 8, 12 y 18. La tabla debe ser creada fuera del procedimiento.

Creacion de Tabla dbo.NumeroPar y Store Procedure que permita insertar números pares del 4 al 22 en una tabla con el nombre dbo.NumeroPar (nro smallint). Excepto los números 8, 12 y 18. La tabla debe ser creada en el procedimiento.

Creacion de Tabla dbo.NumeroPar Store Procedure que permita insertar números pares del 4 al 21 en una tabla con el nombre dbo.NumeroPar (nro smallint). Excepto los números 8, 12 y 18. La tabla debe ser creada fuera del procedimiento. **CORRECTA**

X Creacion de Tabla dbo.NumeroPar y Store Procedure que permita insertar números pares del 4 al 21 en una tabla con el nombre dbo.NumeroPar (nro smallint). Excepto los números 8, 12 y 18. La tabla debe ser creada fuera del procedimiento. **INCORRECTA**

Ninguno de los enunciados ANTERIORES es correcto. Codigo Incorrecto

4) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.

```

CREATE TABLE dbo.NumeroPar (
  nro SMALLINT
);
CREATE PROCEDURE InsertarNumeroPar
  @numero SMALLINT
AS
BEGIN
  IF @numero % 2 <> 0
  BEGIN
    THROW 50001, 'El número ingresado no es par. Inserción fallida.', 1;
  END
  ELSE
  BEGIN
    INSERT INTO dbo.NumeroPar (nro) VALUES (@numero);
  END
END;
EXEC InsertarNumeroPar @numero = 4;

```



sólo se deberá insertar si el número es par. De lo contrario lanzar una excepción. La tabla debe ser creada en el procedimiento.

Creacion de Store Procedure que inserte a la tabla dbo.NumeroPar el número ingresado automaticamente, pero sólo se deberá insertar si el número es par. De lo contrario lanzar una excepción. La tabla debe ser creada fuera del procedimiento.

Creacion de Store Procedure que inserte a la tabla dbo.NumeroPar el número ingresado por parámetro, pero sólo se deberá insertar si el número es par. De lo contrario lanzar una excepción. La tabla debe ser creada fuera del procedimiento. **CORRECTA**

Ninguno de los enunciados ANTERIORES es correcto. Código Incorrecto

2P - Practica - Triggers - CORRECTAS: 1 de 1

1) ¿Qué hace el siguiente trigger INSTEAD OF DELETE en la tabla Employees?

```
CREATE TRIGGER trg_PreventDelete
ON Employees
INSTEAD OF DELETE
AS
BEGIN
    PRINT 'No se permite eliminar empleados de esta tabla.';
END;
```

El trigger permite la eliminación de empleados, pero registra la acción en una tabla de auditoría.

El trigger impide la eliminación de registros en la tabla Employees y muestra un mensaje de error.
CORRECTA

El trigger permite la eliminación de empleados y muestra un mensaje de error después de la eliminación.

El trigger reemplaza el contenido de la tabla Employees por una nueva tabla temporal y muestra un mensaje de error.

Ninguno de los enunciados ANTERIORES es correcto.

2P - Seguridad - General SQLServer - CORRECTAS: 0 de 1

1) Entidades de seguridad y seguridad de objetos de base de datos:

El ámbito de influencia de una entidad de seguridad no depende del ámbito de su definición: Windows, servidor o base de datos; y de si la entidad de seguridad es indivisible o es una colección.

Un Inicio de sesión de Windows es un ejemplo de entidad de seguridad divisible y un Grupo de Windows es un ejemplo de una del tipo colección. **INCORRECTA**



Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta. **CORRECTA**

2P - Seguridad - Usuarios y Esquemas - CORRECTAS: 0 de 2

1) Usuario y esquema dbo:

El usuario dbo es una tabla de seguridad de usuario especial que hay en cada base de datos.

- ✗ Todos los administradores de SQL Server, los miembros del rol fijo de servidor sysdba, el inicio de sesión sa y los propietarios de la base de datos especifican las bases de datos como el usuario dbo. **INCORRECTA**

El usuario dbo tiene todos los permisos en la base de datos master y se limitar o desactivar.

El rol fijo de base de datos db_owner no es lo mismo que la cuenta de usuario que se registra como el propietario de la base de datos. **CORRECTA**

Mas de una respuesta de las ANTERIORES es Correcta.

Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Inicio de sesión sa:

- ✗ La cuenta sa se crea de forma predeterminada cuando se instala una base de datos. **INCORRECTA**

La base de datos predeterminada de la cuenta sa es msdb.

El inicio de sesión, la cuenta sa es miembro del rol fijo de nivel de servidor sysdba.

Este inicio de sesión, la cuenta sa tiene todos los permisos en el servidor y puede limitarse.

Mas de una respuesta de las ANTERIORES es Correcta.

Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta. **CORRECTA**

2P - Seguridad - ROLES - CORRECTAS: 1 de 1

1) Rol público de base de datos y de servidor:

Cada inicio de sesión pertenece al rol fijo de servidor public y cada usuario de base de datos pertenece al rol de base de datos public.

Cuando a un usuario o inicio de sesión no se le han concedido ni denegado permisos concretos para un elemento protegible, hereda los permisos para ese elemento concedidos a public.

El rol fijo de servidor public y el de base de datos public no pueden quitarse, pero se puede revocar los permisos de los roles public.



Todas las respuestas ANTERIORES son correcta. **CORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Seguridad - Elementos protegibles - CORRECTAS: 1 de 1

1) Elementos protegibles => Ámbito protegible: servidor.

Grupo de disponibilidad, Punto de conexión, Iniciar sesión, Rol del servidor, Enlace de servicio remoto.

Grupo de disponibilidad, Clave simétrica, Iniciar sesión, Rol del servidor, Base de datos.

Grupo de disponibilidad, Punto de conexión, Certificate, Rol del servidor, (Base de datos) Rol.

Grupo de disponibilidad, Punto de conexión, Iniciar sesión, Rol del servidor, Base de datos. **CORRECTA**

Mas de una respuesta de las ANTERIORES es Correcta.

Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Seguridad - Jerarquía de Cifrado - CORRECTAS: 0 de 1

1) Jerarquía de cifrado => Conceptos:

Las claves maestras de base de datos se protegen mediante la clave asimétricas maestra de servicio.

La clave maestra de servicio y todas las claves maestras de base de datos son claves asimétricas.

La clave maestra de servicio y todas las claves maestras de base de datos son claves simétricas.

CORRECTA

El módulo de Administración extensible de claves (EKM) mantiene las claves asimétricas o simétricas fuera de SQL Server. **INCORRECTA**

Mas de una respuesta de las ANTERIORES es Correcta.

Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Teoria - Triggers - CORRECTAS: 0 de 2

1) Triggers: Nivel de base de datos

Responden a la necesidad de garantizar la integridad y consistencia de los datos dentro de nuestras tablas de usuario.

No ayudan a mantener las reglas de diseño de nuestra base de datos.

Nos proporcionarán mecanismos para garantizar que nuestra base de datos está diseñada e implementada



TODAS las respuestas ANTERIORES son correctas. **CORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Triggers: Nivel de base de datos

Responden a la necesidad de garantizar la integridad y disponibilidad de los datos dentro de nuestras tablas de usuario.

No ayudan a mantener las reglas de actualización de nuestra base de datos. **INCORRECTA**

Nos proporcionarán mecanismos para garantizar que nuestra base de datos está diseñada e implementada de acuerdo a los estándares de la instalación por defecto.

Tienen dos alcances diferenciados, a nivel de servidor y a nivel de base de datos. **CORRECTA**

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Transacciones y Bloqueos - CORRECTAS: 1 de 2

1) Transacciones Explícitas:

Es aquella en la que se define al final de la transacción para validar la propiedad ACID.

Reciben el nombre de tablas definidas por el usuario.

Se inician y finalizan precisamente mediante comandos como BEGIN TRANSACTION, COMMIT, y ROLLBACK.
CORRECTA

Deben realizarse un control automático de ejecución para cumplir con la propiedad ACID.

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Características ACID: => Aislamiento (Isolation)

Esta propiedad asegura que una transacción se ejecute de manera completa o no se ejecute en absoluto.

Garantiza que una transacción lleve a la base de datos de un estado válido a otro estado válido.

Asegura que las transacciones concurrentes se ejecuten de manera que no afecten entre sí. **CORRECTA**

Garantiza que los cambios realizados por la transacción se guardan de forma que no se perderán, garantizando la integridad de los datos.

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas. **INCORRECTA**



1) Características de los cursos - instrucciones:

DECLARE => define el comportamiento de desplazamiento y la consulta utilizada para generar el conjunto de atributos sobre el que opera el cursor.

OPEN => llena el conjunto de atributos que deben ser analizados para el comportamiento de las sentencias.

FETCH => devuelve el valor de un atributo de la tabla del conjunto de resultados.

CLOSE => libera el conjunto de resultados actual asociado al cursor. **CORRECTA**

Mas de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas. **INCORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Practica - Cursos - CORRECTAS: 1 de 1

1) ¿Qué sentencia es necesaria para abrir un cursor en SQL?

OPEN nombre_del_cursor **CORRECTA**

ACTIVATE nombre_del_cursor

START nombre_del_cursor

EXECUTE nombre_del_cursor

Ninguno de los enunciados ANTERIORES es correcto

RESUMEN

TOTAL CORRECTAS: 14 de 30 **INSUFICIENTE**

2P - Variables SQL	CORRECTAS: 0 de 2
2P - Variables SQL - Características	CORRECTAS: 2 de 2
2P - Procedimientos Almacenados	CORRECTAS: 1 de 2
2P - Procedimientos Almacenados - Características	CORRECTAS: 2 de 2
2P - Procedimientos Almacenados - Ventajas	CORRECTAS: 1 de 1
2P - Procedimientos Almacenados - Parámetros	CORRECTAS: 1 de 2
2P - Tipos de funciones	CORRECTAS: 0 de 2
2P - Práctica - Procedimientos y Funciones	CORRECTAS: 2 de 4
2P - Práctica - Triggers	CORRECTAS: 1 de 1
2P - Seguridad - General SQLServer	CORRECTAS: 0 de 1
2P - Seguridad - Usuarios y Esquemas	CORRECTAS: 0 de 2
2P - Seguridad - ROLES	CORRECTAS: 1 de 1
2P - Seguridad - Elementos protegibles	CORRECTAS: 1 de 1
2P - Seguridad - Jerarquía de Cifrado	CORRECTAS: 0 de 1
2P - Teoría - Triggers	CORRECTAS: 0 de 2



BARRIOS, LUCAS DIEGO

[Mi perfil](#) · [Ayuda](#) · [Cerrar sesión](#)[SALIR](#)UNLaM - DIIT - UEaD. MieL Materias Interactivas en Línea - 2024 - [Aviso legal](#)

Punto de control: Resultado

Estimado/a alumno/a:

Rango de NOTA:

0 a 5 - 1
6 a 14 - 2
15 a 17 - 3
18 a 20 - 4
21 a 22 - 5
23 a 24 - 6
25 a 26 - 7
27 a 28 - 8
29 - 9
30 - 10

2P - Variables SQL - CORRECTAS: 1 de 2

1) Variables SQL:

Permiten almacenar un valor y recuperarlo en sentencias anteriores a ser declarada para emplearlos en otras sentencias.

Son específicas de cada consulta y son ejecutadas automáticamente al realizar la conexión.

Comienzan con "@" (arroba) seguido del nombre (sin espacios), dicho nombre no puede contener cualquier carácter.

Una variable debe ser declarada antes de usarse. **CORRECTA**

 Mas de una respuesta de las ANTERIORES es correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Variables SQL:

Permiten almacenar un valor y recuperarlo más adelante para emplearlos en otras sentencias.

Son específicas de cada consulta y son ejecutadas automáticamente al realizar la conexión.

Comienzan con "@" (arroba) seguido del nombre (sin espacios), dicho nombre no puede contener cualquier carácter.

Una variable debe ser declarada antes de usarse.

 Mas de una respuesta de las ANTERIORES es correcta. **CORRECTA**

1) Variables SQL - Características:

Puede declarar varias variables en una misma sentencia.

No existen variables globales en SQL Server.

Una variable declarada existe dentro del entorno en que se declara.

Una variable a la cual no se le ha asignado un valor contiene "null".

Mas de una respuesta de las ANTERIORES es correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correctas. **CORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Variables SQL - Características:

Solo se puede declarar varias variables en una misma sentencia.

Existen variables globales en SQL Server.

Una variable declarada existe dentro de todo el entorno de trabajo.

Una variable a la cual no se le ha asignado un valor contiene "null". **CORRECTA**

Mas de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Procedimientos Almacenados - CORRECTAS: 0 de 2

1) Procedimientos Almacenados:

Es un grupo de sentencias T-SQL compiladas dentro de un plan de una query.

Son un método de encapsular tareas repetitivas que involucran variables definidas por el usuario para cálculos intermedios. **CORRECTA**

Son variables de control de flujo de ejecución, para la implementación de bloques condicionales o repetitivos.

Son querys y variables que encapsulan código para su reutilización.

Mas de una respuesta de las ANTERIORES es correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Procedimientos Almacenados:

Son querys y variables que encapsulan código para su reutilización.

-  Mas de una respuesta de las ANTERIORES es correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta. **CORRECTA**

2P - Procedimientos Almacenados - Características - CORRECTAS: 2 de 2

1) Procedimientos Almacenados - Características:

No aceptan parámetros de entrada.

Pueden llamar a otros SPs (o sea dentro de un SP se puede llamar a otros SP).

Pueden devolver valores en la forma de parámetros de salida.

No pueden devolver valores en la forma de parámetros de salida.

-  Mas de una respuesta de las ANTERIORES es correcta. **CORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Procedimientos Almacenados - Características:

-  Aceptan parámetros de entrada. **CORRECTA**

Devuelven mas de un valor de retorno (escalar) que indica el éxito o falla de su ejecución.

No pueden llamar a otros SPs (o sea dentro de un SP no se puede llamar a otros SP).

No pueden devolver valores en la forma de parámetros de salida.

Mas de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Procedimientos Almacenados - Ventajas - CORRECTAS: 0 de 1

1) Procedimientos Almacenados - Ventajas:

Encapsulan la lógica de negocio y crean piezas de código reutilizable por la aplicación.

Todas las aplicaciones pueden usar los mismos procedimientos para asegurar un acceso consistente a los datos.

Evitan la exposición de los detalles de las tablas al usuario, haciendo innecesario el acceso a las tablas en



Todas las respuestas ANTERIORES son correctas. **CORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Procedimientos Almacenados - Parametros - CORRECTAS: 2 de 2

1) Procedimientos Almacenados - Parametros:

Cuando se definen parámetros de entrada y salida, estos siempre van finalizados por el signo @, con nombre del parámetro anteponiéndose, y luego una designación del tipo de dato.

Los parámetros de salida deben incluir la palabra clave PUTOUT para diferenciar de los de entrada.

Los Procedimientos nunca pueden incluir parámetros en su definición, de modo tal que pueda crear una lógica de comportamiento más genérica.

- Los parámetros de entrada permiten pasar información al SP. Estos valores son usados como variables locales dentro del procedimiento almacenado. **CORRECTA**

Mas de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Procedimientos Almacenados - Parametros:

Cuando se definen parámetros de entrada y salida, estos siempre van finalizados por el signo @, con nombre del parámetro anteponiéndose, y luego una designación del tipo de dato.

Los parámetros de salida deben incluir la palabra clave OUTPUT para diferenciar de los de entrada.

Los Procedimientos son más flexibles cuando le incluimos parámetros en su definición, de modo tal que pueda crear una lógica de comportamiento más genérica.

Los parámetros de entrada impiden pasar información al SP. Estos valores son ingresados como variables locales al momento de la ejecución dentro del procedimiento almacenado.

- Mas de una respuesta de las ANTERIORES es correcta. **CORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Tipos de funciones - CORRECTAS: 2 de 2

1) Tipos de funciones - Funciones Tabulares MultiSentencias:

Devuelve una tabla que es el resultado de una única sentencia SELECT.

- Devuelve una tabla construida por una o más sentencias Transact-SQL. **CORRECTA**



Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Tipos de funciones - Funciones Escalares:

Devuelve una tabla que es el resultado de de una unica sentencia SELECT.

- Este tipo de funciones es sintacticamente similar a funciones del tipo COUNT(), MAX(). **CORRECTA**

Devuelve una tabla que es el resultado de de una unica sentencia SELECT.

Es similar a una vista ofreciendo mas flexibilidad.

Mas de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Practica - Procedimientos y Funciones - CORRECTAS: 1 de 4

1) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.

```

CREATE PROCEDURE InsertOrUpdateAuto
    @Patente VARCHAR(10),
    @Modelo VARCHAR(50),
    @Conductor VARCHAR(100)
AS
BEGIN
    BEGIN TRY
        INSERT INTO Agencia (Patente, Modelo, Conductor)
        VALUES (@Patente, @Modelo, @Conductor);
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER() = 2627
        BEGIN
            UPDATE Auto
            SET Conductor = @Conductor
            WHERE Patente = @Patente AND Conductor <> @Conductor;
        END
        ELSE
        BEGIN
            DECLARE @ErrorMessage NVARCHAR(5000);
            DECLARE @ErrorSeverity INT;
            DECLARE @ErrorState INT;
            SELECT
                @ErrorMessage = ERROR_MESSAGE(),
                @ErrorSeverity = ERROR_SEVERITY(),
                @ErrorState = ERROR_STATE();
            RAISERROR(@ErrorMessage, @ErrorSeverity, @ErrorState);
        END
    END CATCH
END;

```

Confeccionar un SP que permita insertar un nuevo auto en la tabla Agencia. Si ya existe un registro con la misma Patente, se debe eliminar ese registro y luego insertar el nuevo. Si se produce algún error durante la operación, se debe manejar mostrando un mensaje que indique que se ha eliminado el registro anterior.

Confeccionar un SP que permita insertar un nuevo auto en la tabla. Si ya existe un registro con la misma Patente, se debe insertar un nuevo registro sin actualizar ningún campo. Si se produce un error durante la operación, se debe manejar adecuadamente, mostrando un mensaje que indique que la operación se realizó correctamente.



X Confeccionar un SP que permita insertar un nuevo Auto en la tabla Agencia. Si ya existe un registro con la misma Patente, se debe actualizar solo el campo Conductor si el valor ingresado es diferente al ya cargado. Si se produce algún error durante la operación, se debe manejar. **INCORRECTA**

Ninguno de los enunciados ANTERIORES es correcto. Código Incorrecto **CORRECTA**

2) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.

```
CREATE PROCEDURE dbo.InsertOrUpdateMovie
    @ID INT,
    @Genero VARCHAR(50),
    @Titulo VARCHAR(100),
    @Duracion INT
AS
BEGIN TRY
    INSERT INTO dbo.Movies (ID, Genero, Titulo, Duracion)
    VALUES (@ID, @Genero, @Titulo, @Duracion);
END TRY
BEGIN CATCH
    IF ERROR_NUMBER() = 2627
    BEGIN
        UPDATE dbo.Peliculas
        SET Duracion = @Duracion
        WHERE ID = @ID AND Genero = @Genero;
    END
    ELSE
    BEGIN
        DECLARE @ErrorMessage NVARCHAR(4000);
        DECLARE @ErrorSeverity INT;
        DECLARE @ErrorState INT;
        SELECT
            @ErrorMessage = ERROR_MESSAGE(),
            @ErrorSeverity = ERROR_SEVERITY(),
            @ErrorState = ERROR_STATE();
        RAISERROR(@ErrorMessage, @ErrorSeverity, @ErrorState);
    END
END CATCH
END;
```

X Confeccionar un SP para insertar una nueva película en la tabla. Si ya existe un registro con la misma combinación de ID y Titulo, se debe actualizar el campo Duracion de ese registro. Si se produce algún error durante la operación, se debe manejar. **INCORRECTA**

Confeccionar un SP para insertar una nueva película en la tabla. Si ya existe un registro con la misma combinación de ID y Genero, se debe insertar un nuevo registro sin realizar ninguna actualización. Si se produce un error durante la operación, se debe manejar.

Confeccionar un SP para que permita insertar Peliculas (con el campo ID, Genero, Titulo y Duracion) en una tabla con el nombre dbo.Movies, excepto que el ID y el Genero como combinacion de ambas ya exista en la tabla, en caso de duplicado, actualizar el valor duracion y generar el manejo de error correspondiente.

Confeccionar un SP para insertar una nueva película en la tabla. Si ya existe un registro con la misma combinación de ID y Genero, se debe eliminar el registro existente y luego insertar el nuevo. En caso de error, mostrar mensaje indicando que la película ya existe. En caso de error, se debe mostrar un mensaje indicando que la película ya existe.

Ninguno de los enunciados ANTERIORES es correcto. Código Incorrecto **CORRECTA**

3) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.



```

AS
BEGIN
    -- IF @numero % 2 <> 0
    BEGIN
        THROW 50001, 'El número ingresado no es par. Inserción fallida.', 1;
    END
    ELSE
    BEGIN
        INSERT INTO dbo.NumeroPar (nro) VALUES (@numero);
    END
END;
EXEC InsertarNumeroPar @numero = 4;

```

Creacion de Tabla dbo.NumeroPar y Store Procedure que inserte a la tabla dbo.NumeroPar el número ingresado por parámetro, pero sólo se deberá insertar si el número es par. De lo contrario lanzar una excepción. La tabla debe ser creada en el procedimiento.

Creacion de Tabla dbo.NumeroPar y Store Procedure que inserte a la tabla dbo.NumeroPar el número ingresado automáticamente, pero sólo se deberá insertar si el número es par. De lo contrario lanzar una excepción. La tabla debe ser creada fuera del procedimiento.

Creacion de Tabla dbo.NumeroPar y Store Procedure que inserte a la tabla dbo.NumeroPar el número ingresado automáticamente, pero sólo se deberá insertar si el número es par. De lo contrario lanzar una excepción. La tabla debe ser creada en el procedimiento.

Creacion de Tabla dbo.NumeroPar y Store Procedure que inserte a la tabla dbo.NumeroPar el número ingresado por parámetro, pero sólo se deberá insertar si el número es par. De lo contrario lanzar una excepción. La tabla debe ser creada fuera del procedimiento.



Ninguno de los enunciados ANTERIORES es correcto. Código Incorrecto **CORRECTA**

4) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.

```

CREATE PROCEDURE dbo.InsertarOActualizarCelular
    @Marca NVARCHAR(50),
    @Modelo NVARCHAR(50),
    @Memoria NVARCHAR(20),
    @Procesador NVARCHAR(50)
AS
BEGIN
    SET NOCOUNT ON;
    IF EXISTS (SELECT 1 FROM dbo.Inventariocelulares WHERE Modelo = @Modelo)
    BEGIN
        IF EXISTS (SELECT 1 FROM dbo.Inventariocelulares WHERE Modelo = @Modelo AND Marca <> @Marca)
        BEGIN
            THROW 50002, 'Error: El Modelo ingresado ya existe con una Marca diferente.', 1;
        END
        ELSE
        BEGIN
            UPDATE dbo.Inventario
            SET Memoria = @Memoria,
                Procesador = @Procesador
            WHERE Modelo = @Modelo;
        END
    END
    ELSE
    BEGIN
        INSERT INTO dbo.Inventario (Marca, Modelo, Memoria, Procesador)
        VALUES (@Marca, @Modelo, @Memoria, @Procesador);
    END
END;

```



Confeccionar un SP que permita insertar un nuevo celular en la tabla dbo.Inventariocelulares. Si ya existe cargado el mismo Modelo con Marca diferente, se debe permitir la inserción de un nuevo registro sin generar ningún error. Si el Modelo y la Marca coinciden, se deben actualizar todos los campos del celular (Marca, Memoria, Procesador). En caso de registro, no se debe realizar ninguna acción.

Confeccionar un SP para insertar un nuevo Celular en la tabla dbo.Inventariocelulares. Si ya existe un registro con el mismo Modelo, se debe actualizar solo los campos nuevos del celular con los valores de memoria y procesador, y si el valor ingresado es una Marca y Modelo diferente a los ya cargados se debe cargar el nuevo registro, y si los datos son duplicados se generar el error y manejarlo. **INCORRECTA**

Confeccionar un SP para insertar un nuevo celular en la tabla dbo.Inventariocelulares. Si ya existe un registro con el mismo Modelo, se debe actualizar solo el campo Marca y dejar intactos los valores de Memoria y Procesador. Si se intenta insertar un nuevo registro con un Modelo y Marca duplicados, el procedimiento debe permitir la inserción sin lanzar ningnún error.

Ninguno de los enunciados ANTERIORES es correcto. Código Incorrecto **CORRECTA**

2P - Practica - Triggers - CORRECTAS: 1 de 1

1) ¿Cuál es el resultado de la siguiente instrucción SQL para crear un trigger AFTER DELETE en la tabla Clientes?

```

CREATE TRIGGER trg_EliminarCliente
ON Clientes
AFTER DELETE
AS
BEGIN
    INSERT INTO Cliente_Log (ClienteID, Accion, Fecha)
    SELECT ClienteID, 'Eliminación', GETDATE()
    FROM deleted;
END;

```

El trigger previene la eliminación de clientes en la tabla Clientes.

El trigger actualiza la tabla Clientes cada vez que se elimina un cliente.

El trigger registra todas las acciones que ocurren en la tabla Clientes, independientemente de si son eliminaciones, inserciones o actualizaciones, insertando el ID del cliente, la acción y la fecha.

El trigger registra las eliminaciones de clientes en la tabla Cliente_Log, insertando el ID del cliente eliminado, la acción y la fecha de la eliminación. **CORRECTA**

Ninguno de los enunciados ANTERIORES es correcto.

2P - Seguridad - General SQLServer - CORRECTAS: 1 de 1



Toda entidad de seguridad tiene un identificador de seguridad (PID).

- El ámbito de influencia de una entidad de seguridad depende del ámbito de su definición: Windows, servidor o base de datos; y de si la entidad de seguridad es indivisible o es una colección. **CORRECTA**

Este tema se aplica a todas las versiones de SQL Server, y no hay algunas restricciones en las entidades de seguridad a nivel de servidor de SQL Database.

Mas de una respuesta de las ANTERIORES es Correcta.

Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Seguridad - Usuarios y Esquemas - CORRECTAS: 0 de 2

1) Inicio de sesión sa:

El inicio de sesión de SQL Server, la cuenta sa es una entidad de seguridad a nivel de servidor.

La cuenta sa se crea de forma predeterminada cuando se instala una instancia.

El inicio de sesión sa es miembro del rol fijo de nivel de servidor sysadmin.

Este inicio de sesión, la cuenta sa tiene todos los permisos en el servidor y no puede limitarse.

- Mas de una respuesta de las ANTERIORES es Correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correcta. **CORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Usuario y esquema dbo:

El usuario dbo es una tabla de seguridad de usuario especial que hay en cada base de datos.

Todos los administradores de SQL Server, los miembros del rol fijo de servidor sysdba, el inicio de sesión sa y los propietarios de la base de datos especifican las bases de datos como el usuario dbo.

El usuario dbo tiene todos los permisos en la base de datos master y se limitar o desactivar.

El rol fijo de base de datos db_owner no es lo mismo que la cuenta de usuario que se registra como el propietario de la base de datos. **CORRECTA**

- Mas de una respuesta de las ANTERIORES es Correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Seguridad - ROLES - CORRECTAS: 0 de 1

1) Rol público de base de datos y de servidor:

El rol fijo de servidor public y el de base de datos public no pueden quitarse, pero se puede revocar los permisos de los roles public.

Hay muchos de los permisos que se asignan a los roles public de forma predeterminada. La mayoría de estos permisos son necesarios para realizar operaciones rutinarias en la base de datos; el tipo de tareas que todos los usuarios deben poder hacer.

Mas de una respuesta de las ANTERIORES es Correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correcta. **CORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Seguridad - Elementos protegibles - CORRECTAS: 0 de 1

1) Elementos protegibles => Ámbito protegible: esquema OBJETOS:

Esquemas XML, Agregada, Función, Ver, Tabla externa.

Agregada, Función, Ver, Tabla externa. **CORRECTA**

Procedimiento, Tipo, Cola, Synonym, Usuario.

Procedimiento, Tipo, Cola, Synonym, Tabla.

Mas de una respuesta de las ANTERIORES es Correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Seguridad - Jerarquía de Cifrado - CORRECTAS: 0 de 1

1) Jerarquía de cifrado => Conceptos:

Las claves maestras de base de datos se protegen mediante la clave asimétricas maestra de servicio.

La clave maestra de servicio y todas las claves maestras de base de datos son claves asimétricas.

La clave maestra de servicio y todas las claves maestras de base de datos son claves simétricas.

CORRECTA

El módulo de Administración extensible de claves (EKM) mantiene las claves asimétricas o simétricas fuera de SQL Server.

Mas de una respuesta de las ANTERIORES es Correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Teoria - Triggers - CORRECTAS: 1 de 2



No ayudan a mantener las reglas de actualización de nuestra base de datos.

Nos proporcionarán mecanismos para garantizar que nuestra base de datos está diseñada e implementada de acuerdo a los estándares de la instalación por defecto.

A esta categoría de eventos pertenecen entre otros los de CREATE / ALTER /DROP View - CREATE / ALTER /DROP Table - CREATE / ALTER /DROP Schema **CORRECTA**

Mas de una respuesta de las ANTERIORES es correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Triggers:

Los TRIGGERS son muy similares en su concepto a los procedimientos almacenados, son código Transact-SQL similares en código y ejecución.

Un TRIGGER se ejecuta en respuesta a un evento.

La manipulación de datos son del tipo TRIGGERS DMZ o por la manipulación de esquemas como los TRIGGERS DLL.

Reciben parámetros de una forma muy especial, en forma de unas tablas virtuales, llamadas inserted y deleted.

Mas de una respuesta de las ANTERIORES es correcta. **CORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Transacciones y Bloqueos - CORRECTAS: 0 de 2

1) Características ACID: => Atomicidad (Atomicity)

Asegura que las transacciones concurrentes se ejecuten de manera que no afecten entre sí. **INCORRECTA**

Garantiza que los cambios realizados por la transacción se guardan de forma que no se perderán, garantizando la integridad de los datos.

Esta propiedad asegura que una transacción se ejecute de manera completa o no se ejecute en absoluto. **CORRECTA**

Garantiza que una transacción lleve a la base de datos de un estado válido a otro estado válido.

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Transacciones Explícitas:



Deben realizarse un control automatico de ejecucion para cumplir con la propiedad ACID.

- Mas de una respuesta de las ANTERIORES es correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Teoria - Cursos - CORRECTAS: 0 de 1

1) Características de los cursos:

Nos permite movernos por los registros de una consulta como un único bloque para analizar el resultado todo junto.

Nos permite recorrer los registros del resultado obtenido de una sentencia WHERE.

Se usa para realizar una acción si el código identificador de un registro no es autonumérico.

Su uso permiten procesar el conjunto de resultados de fila en fila. **CORRECTA**

- Mas de una respuesta de las ANTERIORES es correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Practica - Cursos - CORRECTAS: 1 de 1

1) ¿Qué sentencia es necesaria para abrir un cursor en SQL?

- OPEN nombre_del_cursor **CORRECTA**

ACTIVATE nombre_del_cursor

START nombre_del_cursor

EXECUTE nombre_del_cursor

Ninguno de los enunciados ANTERIORES es correcto

RESUMEN

TOTAL CORRECTAS: 13 de 30 **INSUFICIENTE**

2P - Variables SQL CORRECTAS: 1 de 2

2P - Variables SQL - Caracteristicas CORRECTAS: 1 de 2

2P - Procedimientos Almacenados CORRECTAS: 0 de 2

2P - Procedimientos Almacenados - Caracteristicas CORRECTAS: 2 de 2

2P - Procedimientos Almacenados - Ventajas CORRECTAS: 0 de 1

2P - Procedimientos Almacenados - Parametros CORRECTAS: 2 de 2



GUARNIERI, JULIAN EZEQUIEL

[Mi perfil](#) · [Ayuda](#) · [Cerrar sesión](#)

2P - Seguridad - ROLES	CORRECTAS: 0 de 1
2P - Seguridad - Elementos protegibles	CORRECTAS: 0 de 1
2P - Seguridad - Jerarquía de Cifrado	CORRECTAS: 0 de 1
2P - Teoria - Triggers	CORRECTAS: 1 de 2
2P - Transacciones y Bloqueos	CORRECTAS: 0 de 2
2P - Teoria - Cursores	CORRECTAS: 0 de 1
2P - Practica - Cursores	CORRECTAS: 1 de 1

[SALIR](#)

Punto de control: Resultado

Estimado/a alumno/a:

Rango de NOTA:

0 a 5 - 1
6 a 14 - 2
15 a 17 - 3
18 a 20 - 4
21 a 22 - 5
23 a 24 - 6
25 a 26 - 7
27 a 28 - 8
29 - 9
30 - 10

2P - Variables SQL - CORRECTAS: 0 de 2

1) Variables SQL:

Permiten almacenar un valor y recuperarlo en sentencias anteriores a ser declarada para emplearlos en otras sentencias.

Son específicas de cada consulta y son ejecutadas automáticamente al realizar la conexión.

 Comienzan con "@" (arroba) seguido del nombre (sin espacios), dicho nombre no puede contener cualquier carácter. **INCORRECTA**

Una variable debe ser declarada al final para usarse en el código anterior.

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta. **CORRECTA**

2) Variables SQL:

Permiten almacenar un valor y recuperarlo en sentencias anteriores a ser declarada para emplearlos en otras sentencias.

Son específicas de cada consulta y son ejecutadas automáticamente al realizar la conexión.

Comienzan con "@" (arroba) seguido del nombre (sin espacios), dicho nombre puede contener cualquier carácter. **CORRECTA**

Una variable debe ser declarada al final para usarse en el código anterior.

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

1) Variables SQL - Características:

Solo se puede declarar varias variables en una misma sentencia.

Existen variables globales en SQL Server.

Una variable declarada existe dentro de todo el entorno de trabajo.

-  Una variable a la cual no se le ha asignado un valor contiene "null". **CORRECTA**

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Variables SQL - Características:

Solo se puede declarar varias variables en una misma sentencia.

Existen variables globales en SQL Server.

-  Una variable declarada existe dentro del entorno en que se declara. **CORRECTA**

Una variable a la cual no se le ha asignado un valor contiene "0" (Cero).

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Procedimientos Almacenados - CORRECTAS: 1 de 2

1) Procedimientos Almacenados:

Es un grupo de sentencias T-SQL compiladas dentro de un plan de una query.

Son un método de encapsular tareas repetitivas que no involucran variables definidas por el motor de SQL para cálculos intermedios.

Son variables de control de flujo de ejecución, para la implementación de bloques condicionales o repetitivos.

Son querys y variables que encapsulan código para su reutilización.

-  Mas de una respuesta de las ANTERIORES es correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta. **CORRECTA**

2) Procedimientos Almacenados:

Es un grupo de sentencias T-SQL compiladas dentro de un plan de ejecución.

Son un método de encapsular tareas repetitivas que involucran variables definidas por el usuario para cálculos intermedios.

 Todas las respuestas ANTERIORES son correctas. **CORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Procedimientos Almacenados - Características - CORRECTAS: 2 de 2

1) Procedimientos Almacenados - Características:

 Aceptan parámetros de entrada. **CORRECTA**

Devuelven mas de un valor de retorno (escalar) que indica el éxito o falla de su ejecución.

No pueden llamar a otros SPs (o sea dentro de un SP no se puede llamar a otros SP).

No pueden devolver valores en la forma de parámetros de salida.

Mas de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Procedimientos Almacenados - Características:

No aceptan parámetros de entrada.

 Devuelven un valor de retorno (escalar) que indica el éxito o falla de su ejecución. **CORRECTA**

No pueden llamar a otros SPs (o sea dentro de un SP no se puede llamar a otros SP).

No pueden devolver valores en la forma de parámetros de salida.

Mas de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Procedimientos Almacenados - Ventajas - CORRECTAS: 0 de 1

1) Procedimientos Almacenados - Ventajas:

Encapsulan la lógica de negocio y crean piezas de código reutilizable por la aplicación.

Todas las aplicaciones pueden usar los mismos procedimientos para asegurar un acceso consistente a los datos.

Evitan la exposición de los detalles de las tablas al usuario, haciendo innecesario el acceso a las tablas en forma directa, lo que incrementa sensiblemente la seguridad.

Puede otorgarse permisos de ejecución a un procedimiento a un usuario aun cuando no tenga permisos sobre las tablas o vistas usadas por el procedimiento.

 Mas de una respuesta de las ANTERIORES es correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correctas. **CORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

Cuando se definen parámetros de entrada y salida, estos siempre van finalizados por el signo @, con nombre del parámetro anteponiéndose, y luego una designación del tipo de dato.

- Los parámetros de salida deben incluir la palabra clave OUTPUT para diferenciar de los de entrada. **CORRECTA**

Los Procedimientos nunca pueden incluir parámetros en su definición, de modo tal que pueda crear una lógica de comportamiento más genérica.

Los parámetros de entrada impiden pasar información al SP. Estos valores son ingresados como variables locales al momento de la ejecución dentro del procedimiento almacenado.

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Procedimientos Almacenados - Parámetros:

Cuando se definen parámetros de entrada y salida, estos siempre van finalizados por el signo @, con nombre del parámetro anteponiéndose, y luego una designación del tipo de dato.

Los parámetros de salida deben incluir la palabra clave PUTOUT para diferenciar de los de entrada.

- Los Procedimientos son más flexibles cuando le incluimos parámetros en su definición, de modo tal que pueda crear una lógica de comportamiento más genérica. **CORRECTA**

Los parámetros de entrada impiden pasar información al SP. Estos valores son ingresados como variables locales al momento de la ejecución dentro del procedimiento almacenado.

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Tipos de funciones - CORRECTAS: 2 de 2

1) Tipos de funciones - Funciones Escalares:

Devuelven un solo tipo de valor definido en la cláusula "Returns".

Este tipo de funciones es sintácticamente similar a funciones del tipo COUNT(), MAX().

Devuelve una tabla que es el resultado de una única sentencia SELECT.

Es similar a una vista ofreciendo más flexibilidad.

- Más de una respuesta de las ANTERIORES es correcta. **CORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Tipos de funciones - Funciones Tabulares en Línea:

Devuelve una tabla que es el resultado de una única sentencia SELECT.

Devuelven un solo tipo de valor definido en la cláusula "Returns".



Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Practica - Procedimientos y Funciones - CORRECTAS: 4 de 4

- 1) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.

```

CREATE PROCEDURE dbo.InsertOrUpdateMovie
    @ID INT,
    @Genero VARCHAR(50),
    @Titulo VARCHAR(100),
    @Duracion INT
AS
BEGIN TRY
    INSERT INTO dbo.Movies (ID, Genero, Titulo, Duracion)
    VALUES (@ID, @Genero, @Titulo, @Duracion);
END TRY
BEGIN CATCH
    IF ERROR_NUMBER() = 2627
    BEGIN
        UPDATE dbo.Peliculas
        SET Duracion = @Duracion
        WHERE ID = @ID AND Genero = @Genero;
    END
    ELSE
    BEGIN
        DECLARE @ErrorMessage NVARCHAR(4000);
        DECLARE @ErrorSeverity INT;
        DECLARE @ErrorState INT;
        SELECT
            @ErrorMessage = ERROR_MESSAGE(),
            @ErrorSeverity = ERROR_SEVERITY(),
            @ErrorState = ERROR_STATE();
        RAISERROR(@ErrorMessage, @ErrorSeverity, @ErrorState);
    END
END CATCH
END;

```

Confeccionar un SP para insertar una nueva película en la tabla. Si ya existe un registro con la misma combinación de ID y Titulo, se debe actualizar el campo Duracion de ese registro. Si se produce algún error durante la operación, se debe manejar.

Confeccionar un SP para insertar una nueva película en la tabla. Si ya existe un registro con la misma combinación de ID y Genero, se debe insertar un nuevo registro sin realizar ninguna actualización. Si se produce un error durante la operación, se debe manejar.

Confeccionar un SP para que permita insertar Peliculas (con el campo ID, Genero, Titulo y Duracion) en una tabla con el nombre dbo.Movies, excepto que el ID y el Genero como combinacion de ambas ya exista en la tabla, en caso de duplicado, actualizar el valor duracion y generar el manejo de error correspondiente.

Confeccionar un SP para insertar una nueva película en la tabla. Si ya existe un registro con la misma combinación de ID y Genero, se debe eliminar el registro existente y luego insertar el nuevo. En caso de error, mostrar mensaje indicando que la película ya existe. En caso de error, se debe mostrar un mensaje indicando que la película ya existe.



Ninguno de los enunciados ANTERIORES es correcto. Código Incorrecto **CORRECTA**

- 2) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.



```

IF EXISTS (SELECT * FROM dbo.alumnos WHERE DNI = @DNI)
BEGIN
    IF EXISTS (SELECT 1 FROM dbo.alumnos WHERE DNI = @DNI AND NombreApellido <> @NombreApellido)
    BEGIN
        THROW 50001, 'Error: El DNI ingresado ya existe con un nombre-apellido diferente.', 1;
    END
    ELSE
    BEGIN
        UPDATE dbo.alumnos
        SET Celular = @Celular,
            Email = @Email
        WHERE DNI = @DNI;
    END
END
ELSE
BEGIN
    INSERT INTO dbo.alumnos (DNI, NombreApellido, Celular, Email)
    VALUES (@DNI, @NombreApellido, @Celular, @Email);
END
END;

```

Confeccionar un SP que permita insertar una nueva persona en la tabla dbo.alumnos. Si ya existe un registro con la misma persona, se debe actualizar los campos nuevos del alumno, y si los valores ingresados son identicos a los del alumno ya cargado, se debe generar el error correspondiente y manejarlo.

Confeccionar un SP que permita insertar una nueva persona en la tabla dbo.estudiantes. Si ya existe un registro con el mismo DNI, se debe actualizar el nombre, teléfono y correo electrónico. En caso de intentar insertar una persona con un DNI que ya existe, pero con un nombre distinto, debe actualizar también el nombre sin generar ningún error.

Confeccionar un SP que permita insertar una nueva persona en la tabla dbo.personas. Si ya existe un registro con el mismo DNI y el mismo nombre_completo, debe actualizar únicamente el campo email (sin tocar el celular). Si el DNI existe pero el nombre_completo es diferente, debe permitir insertar un nuevo registro duplicado con el mismo DNI y el nuevo nombre_completo.

Confeccionar un SP que permita insertar una nueva persona en la tabla dbo.alumnos. Si ya existe un registro con la misma persona, se debe actualizar solo los campos nuevos del celular y email, y si el valor ingresado es un DNI igual y nombre-apellido diferentes a uno ya cargado se debe generar el error correspondiente y manejarlo. **CORRECTA**

Ninguno de los enunciados ANTERIORES es correcto. Código Incorrecto

3) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.

```

CREATE TABLE dbo.NumeroPar (
    nro SMALLINT
);
CREATE PROCEDURE InsertarNumeroPar
    @numero SMALLINT
AS
BEGIN
    -- IF @numero % 2 <> 0
    BEGIN
        THROW 50001, 'El número ingresado no es par. Inserción fallida.', 1;
    END
    ELSE
    BEGIN
        INSERT INTO dbo.NumeroPar (nro) VALUES (@numero);
    END
END;
EXEC InsertarNumeroPar @numero = 4;

```

ingresado automaticamente, pero sólo se deberá insertar si el número es par. De lo contrario lanzar una excepción. La tabla debe ser creada fuera del procedimiento.

Creacion de Tabla dbo.NumeroPar y Store Procedure que inserte a la tabla dbo.NumeroPar el número ingresado automaticamente, pero sólo se deberá insertar si el número es par. De lo contrario lanzar una excepción. La tabla debe ser creada en el procedimiento.

Creacion de Tabla dbo.NumeroPar y Store Procedure que inserte a la tabla dbo.NumeroPar el número ingresado por parámetro, pero sólo se deberá insertar si el número es par. De lo contrario lanzar una excepción. La tabla debe ser creada fuera del procedimiento.



Ninguno de los enunciados ANTERIORES es correcto. Código Incorrecto **CORRECTA**

4) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.

```

CREATE TABLE dbo.Clientes (
    DNI VARCHAR(10) PRIMARY KEY,
    Apellido NVARCHAR(100),
    Nombre NVARCHAR(100)
);
CREATE PROCEDURE p_InsClientes
    @dni VARCHAR(10),
    @apellido NVARCHAR(100),
    @nombre NVARCHAR(100)
AS
BEGIN
    IF EXISTS (SELECT 1 FROM dbo.Clientes WHERE DNI = @dni)
    BEGIN
        THROW 50001, 'El DNI ingresado ya existe. Inserción fallida.', 1;
    END
    ELSE
    BEGIN
        INSERT INTO dbo.Clientes (DNI, Apellido, Nombre)
        VALUES (@dni, @apellido, @nombre);
    END
END;
EXEC p_InsClientes @dni = '12345678A', @apellido = 'Pérez', @nombre = 'Maria';

```

Confeccionar la creacion de tabla dbo.Clientes y el Store procedure que permita insertar Personas (con el campo DNI, Apellido y Nombre) en una tabla con el nombre dbo.Clientes, excepto que el DNI ya exista en la tabla se actualiza el Apellido y Nombre en caso de duplicado, y generar el manejo de error correspondiente.

Confeccionar la creacion de tabla dbo.Clientes y el Store procedure que permita insertar Personas (con el campo DNI, Apellido y Nombre) en una tabla con el nombre dbo.Clientes, en caso de que el DNI ya exista, simplemente lo ignora y no genera ningún error.



Confeccionar la creacion de tabla dbo.Clientes y el Store procedure que permita insertar Personas (con el campo DNI, Apellido y Nombre) en una tabla con el nombre dbo.Clientes, excepto que el DNI ya exista en la tabla. En caso de que exista la persona a cargar, generar el manejo de error correspondiente. **CORRECTA**

Confeccionar la creacion de tabla dbo.Clientes y el Store procedure que permita insertar Personas (con el campo DNI, Apellido y Nombre) en una tabla con el nombre dbo.Cliente, excepto que el DNI ya exista en la tabla. En caso de que exista la persona a cargar, generar el manejo de error correspondiente.

Ninguno de los enunciados ANTERIORES es correcto. Código Incorrecto

2P - Practica - Triggers - CORRECTAS: 1 de 1

1) ¿Qué hace el siguiente trigger AFTER INSERT en la tabla Orders?



```

AS
BEGIN
    INSERT INTO OrderHistory (OrderID, Status, ChangeDate)
    SELECT OrderID, 'Nuevo', GETDATE()
    FROM inserted;
END;

```

El trigger actualiza la columna Status en la tabla Orders después de cada inserción.

El trigger bloquea cualquier inserción en la tabla Orders.

- El trigger inserta un nuevo registro en la tabla OrderHistory cada vez que se inserta un nuevo pedido en Orders. **CORRECTA**

El trigger elimina el registro recién insertado en la tabla Orders.

Ninguno de los enunciados ANTERIORES es correcto.

2P - Seguridad - General SQLServer - CORRECTAS: 0 de 1

- 1) SQL Server admite los siguientes tipos de inicios de sesión:

SQL Server usa Windows para autenticar cuentas de usuario de Windows ya sea de Active Directory.

SQL Server usa usuarios locales del servidor donde está instalado el motor de Base de Datos, y conceder acceso a un grupo de Windows otorga acceso a todos los inicios de sesión de usuario de Windows que son miembros del grupo.

SQL Server almacena el nombre de usuario y un hash de la contraseña en la base de datos master.

SQL Server admite usuarios de base de datos independientes para la autenticación de Windows y SQL Server.

- Mas de una respuesta de las ANTERIORES es Correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correcta. **CORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Seguridad - Usuarios y Esquemas - CORRECTAS: 1 de 2

- 1) Inicio de sesión sa:

La cuenta sa se crea de forma predeterminada cuando se instala una base de datos.

La base de datos predeterminada de la cuenta sa es msdb.

El inicio de sesión, la cuenta sa es miembro del rol fijo de nivel de servidor sysdba.

El inicio de sesión de SQL Server, la cuenta sa es una entidad de seguridad a nivel de servidor.

CORRECTA

- Mas de una respuesta de las ANTERIORES es Correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correcta.

TODOS LOS administradores de SQL Server, los miembros del rol fijo de servidor sysadmin, el inicio de sesión sa y los propietarios de la base de datos especifican las bases de datos como el usuario dbo.

El usuario dbo tiene todos los permisos en la base de datos master y se limitar o desactivar.

La cuenta de usuario dbo es lo mismo que el rol fijo de base de datos db_owner.

El rol fijo de base de datos db_owner es lo mismo que la cuenta de usuario que se registra como el propietario de la base de datos.

Mas de una respuesta de las ANTERIORES es Correcta.

Todas las respuestas ANTERIORES son correcta.

- Ninguna de TODAS las respuestas ANTERIORES es correcta. **CORRECTA**

2P - Seguridad - ROLES - CORRECTAS: 0 de 1

1) Rol público de base de datos y de servidor:

Cada inicio de sesión pertenece al rol fijo de servidor public y cada usuario de base de datos pertenece al rol de base de datos public.

Cuando a un usuario o inicio de sesión no se le han concedido ni denegado permisos concretos para un elemento protegible, hereda los permisos para ese elemento concedidos a public.

El rol fijo de servidor public y el de base de datos public no pueden quitarse, pero se puede revocar los permisos de los roles public.

Hay muchos de los permisos que se asignan a los roles public de forma predeterminada. La mayoría de estos permisos son necesarios para realizar operaciones rutinarias en la base de datos; el tipo de tareas que todos los usuarios deben poder hacer.

Mas de una respuesta de las ANTERIORES es Correcta.

Todas las respuestas ANTERIORES son correcta. **CORRECTA**

- Ninguna de TODAS las respuestas ANTERIORES es correcta. **INCORRECTA**

2P - Seguridad - Elementos protegibles - CORRECTAS: 0 de 1

1) Elementos protegibles => Ámbito protegible: base de datos

Grupo de disponibilidad, Punto de conexión, Iniciar sesión, Rol del servidor, Base de datos.

- Rol de aplicación, Punto de conexión, Clave asimétrica, Certificate, Contrato. **INCORRECTA**

Ruta, Esquema, Lista de propiedades de búsqueda, Service, Clave simétrica. **CORRECTA**

Catálogo de texto completo, Punto de conexión, Tipo de mensaje, Enlace de servicio remoto, (Base de datos) Rol.

Mas de una respuesta de las ANTERIORES es Correcta.

Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta.



LAS CLAVES MAESTRAS DE BASE DE DATOS SE PROTEGEN MEDIANTE LA CLAVE ASIMÉTRICA MAESTRA DE SERVICIO.

La clave maestra de servicio y todas las claves maestras de base de datos son claves asimétricas.

La clave maestra de servicio y todas las claves maestras de base de datos son claves simétricas.

CORRECTA

-  El módulo de Administración extensible de claves (EKM) mantiene las claves asimétricas o simétricas fuera de SQL Server. **INCORRECTA**

Mas de una respuesta de las ANTERIORES es Correcta.

Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Teoria - Triggers - CORRECTAS: 0 de 2

1) Triggers:

Los TRIGGERS son muy similares en su concepto a los procedimientos almacenados, son código Transact-SQL similares en código y ejecución.

-  Un TRIGGER se ejecuta en respuesta a un evento. **INCORRECTA**

La manipulación de datos son del tipo TRIGGERS DMZ o por la manipulación de esquemas como los TRIGGERS DLL.

Reciben parámetros de una forma muy especial, en forma de unas tablas virtuales, llamadas inserted y deleted.

Mas de una respuesta de las ANTERIORES es correcta. **CORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Triggers: Nivel de servidor

Responden a los eventos que son propios de la base de datos y no a los que están en el alcance de la instancia.

A esta categoría de eventos pertenecen entre otros los de CREATE LOGIN o CREATE/ALTER/DROP Database y los relativos por ejemplo a los nuevos ENDPOINTS. **CORRECTA**

Están muy alineadas con las de los que son a nivel de Instancia e incluso la información.

La Cláusula ON ALL SERVER dentro de una instrucción CREATE TRIGGER no afectaría a todo el servidor sino a la Base de Datos especificada.

-  Mas de una respuesta de las ANTERIORES es correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Transacciones y Bloqueos - CORRECTAS: 1 de 2

1) Transacciones Implícitas:

Genera una cadena fraccionada de transacciones.

× Mas de una respuesta de las ANTERIORES es correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Características ACID: => Atomicidad (Atomicity)

Asegura que las transacciones concurrentes se ejecuten de manera que no afecten entre sí.

Garantiza que los cambios realizados por la transacción se guardan de forma que no se perderán, garantizando la integridad de los datos.

✓ Esta propiedad asegura que una transacción se ejecute de manera completa o no se ejecute en absoluto. **CORRECTA**

Garantiza que una transacción lleve a la base de datos de un estado válido a otro estado válido.

Mas de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Teoria - Cursos - CORRECTAS: 0 de 1

1) Características de los cursos - instrucciones:

DECLARE => define el comportamiento de desplazamiento y la consulta utilizada para generar el conjunto de resultados sobre el que opera el cursor.

OPEN => llena el conjunto de resultados.

× FETCH => devuelve una fila del conjunto de resultados. **INCORRECTA**

CLOSE => libera el conjunto de resultados actual asociado al cursor.

Mas de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas. **CORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Practica - Cursos - CORRECTAS: 0 de 1

1) ¿Cuál es la sentencia correcta para liberar la memoria de un cursor después de utilizarlo?

FREE nombre_del_cursor

× CLOSE nombre_del_cursor **INCORRECTA**

SHUTDOWN nombre_del_cursor

DEALLOCATE nombre_del_cursor **CORRECTA**

Ninguna de las opciones ANTERIORES es correcta.



2P - Variables SQL - Características	CORRECTAS: 2 de 2
2P - Procedimientos Almacenados	CORRECTAS: 1 de 2
2P - Procedimientos Almacenados - Características	CORRECTAS: 2 de 2
2P - Procedimientos Almacenados - Ventajas	CORRECTAS: 0 de 1
2P - Procedimientos Almacenados - Parámetros	CORRECTAS: 2 de 2
2P - Tipos de funciones	CORRECTAS: 2 de 2
2P - Práctica - Procedimientos y Funciones	CORRECTAS: 4 de 4
2P - Práctica - Triggers	CORRECTAS: 1 de 1
2P - Seguridad - General SQLServer	CORRECTAS: 0 de 1
2P - Seguridad - Usuarios y Esquemas	CORRECTAS: 1 de 2
2P - Seguridad - ROLES	CORRECTAS: 0 de 1
2P - Seguridad - Elementos protegibles	CORRECTAS: 0 de 1
2P - Seguridad - Jerarquía de Cifrado	CORRECTAS: 0 de 1
2P - Teoría - Triggers	CORRECTAS: 0 de 2
2P - Transacciones y Bloqueos	CORRECTAS: 1 de 2
2P - Teoría - Cursos	CORRECTAS: 0 de 1
2P - Práctica - Cursos	CORRECTAS: 0 de 1

[SALIR](#)



Punto de control: Resultado

Estimado/a alumno/a:

Rango de NOTA:

0 a 5 - 1
6 a 14 - 2
15 a 17 - 3
18 a 20 - 4
21 a 22 - 5
23 a 24 - 6
25 a 26 - 7
27 a 28 - 8
29 - 9
30 - 10

2P - Variables SQL - CORRECTAS: 0 de 2

1) Variables SQL:

Permiten almacenar un valor y recuperarlo en sentencias anteriores a ser declarada para emplearlos en otras sentencias.

Son específicas de cada conexión y son liberadas automáticamente al abandonar la conexión.

Comienzan con "@" (arroba) seguido del nombre (sin espacios), dicho nombre puede contener cualquier carácter.

Una variable debe ser declarada al final para usarse en el código anterior.

Más de una respuesta de las ANTERIORES es correcta. **CORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta. **INCORRECTA**

2) Variables SQL:

Permiten almacenar un valor y recuperarlo en sentencias anteriores a ser declarada para emplearlos en otras sentencias.

Son específicas de cada consulta y son ejecutadas automáticamente al realizar la conexión.

Comienzan con "@" (arroba) seguido del nombre (sin espacios), dicho nombre no puede contener ningún carácter.

Una variable debe ser declarada antes de usarse. **CORRECTA**



2P - Variables SQL - Características - CORRECTAS: 0 de 2

1) Variables SQL - Características:

Puede declarar varias variables en una misma sentencia.

No existen variables globales en SQL Server.

Una variable declarada existe dentro del entorno en que se declara.

Una variable a la cual no se le ha asignado un valor contiene "null". **INCORRECTA**

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas. **CORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Variables SQL - Características:

Solo se puede declarar varias variables en una misma sentencia.

Existen variables globales en SQL Server.

Una variable declarada existe dentro del entorno en que se declara. **CORRECTA**

Una variable a la cual no se le ha asignado un valor contiene "0" (Cero).

Más de una respuesta de las ANTERIORES es correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Procedimientos Almacenados - CORRECTAS: 0 de 2

1) Procedimientos Almacenados:

Es un grupo de sentencias T-SQL compiladas dentro de un plan de una query.

Son un método de encapsular tareas repetitivas que no involucran variables definidas por el motor de SQL para cálculos intermedios.

Son sentencias de control de flujo de ejecución, para la implementación de bloques condicionales o repetitivos.

Son módulos o rutinas que encapsulan código para su reutilización.

Más de una respuesta de las ANTERIORES es correcta. **CORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta. **INCORRECTA**

2) Procedimientos Almacenados:



Son querys y variables que encapsulan código para su reutilización.

- Mas de una respuesta de las ANTERIORES es correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta. **CORRECTA**

2P - Procedimientos Almacenados - Características - CORRECTAS: 1 de 2

1) Procedimientos Almacenados - Características:

Aceptan parámetros de entrada.

Devuelven un valor de retorno (escalar) que indica el éxito o falla de su ejecución.

Pueden llamar a otros SPs (o sea dentro de un SP se puede llamar a otros SP).

Pueden devolver valores en la forma de parámetros de salida.

Mas de una respuesta de las ANTERIORES es correcta.

- Todas las respuestas ANTERIORES son correctas. **CORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Procedimientos Almacenados - Características:

No aceptan parámetros de entrada.

Devuelven mas de un valor de retorno (escalar) que indica el éxito o falla de su ejecución.

Pueden llamar a otros SPs (o sea dentro de un SP se puede llamar a otros SP). **CORRECTA**

No pueden devolver valores en la forma de parámetros de salida.

- Mas de una respuesta de las ANTERIORES es correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Procedimientos Almacenados - Ventajas - CORRECTAS: 0 de 1

1) Procedimientos Almacenados - Ventajas:

Encapsulan la lógica de negocio y crean piezas de código reutilizable por la aplicación.

Todas las aplicaciones pueden usar los mismos procedimientos para asegurar un acceso consistente a los datos.

Evitan la exposición de los detalles de las tablas al usuario, haciendo innecesario el acceso a las tablas en



Todas las respuestas ANTERIORES son correctas. **CORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Procedimientos Almacenados - Parametros - CORRECTAS: 2 de 2

1) Procedimientos Almacenados - Parametros:

Cuando se definen parámetros de entrada y salida, estos siempre van finalizados por el signo @, con nombre del parámetro anteponiéndose, y luego una designación del tipo de dato.

Los parámetros de salida deben incluir la palabra clave PUTOUT para diferenciar de los de entrada.

Los Procedimientos son más flexibles cuando le incluimos parámetros en su definición, de modo tal que pueda crear una lógica de comportamiento más genérica.

Los parámetros de entrada permiten pasar información al SP. Estos valores son usados como variables locales dentro del procedimiento almacenado.

Mas de una respuesta de las ANTERIORES es correcta. **CORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Procedimientos Almacenados - Parametros:

Cuando se definen parámetros de entrada y salida, estos siempre van precedidos por el signo @, seguido del nombre del parámetro y luego una designación del tipo de dato.

Los parámetros de salida deben incluir la palabra clave OUTPUT para diferenciar de los de entrada.

Los Procedimientos son más flexibles cuando le incluimos parámetros en su definición, de modo tal que pueda crear una lógica de comportamiento más genérica.

Los parámetros de entrada permiten pasar información al SP. Estos valores son usados como variables locales dentro del procedimiento almacenado.

Mas de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas. **CORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Tipos de funciones - CORRECTAS: 0 de 2

1) Tipos de funciones - Funciones Escalares:

Devuelve una tabla que es el resultado de una única sentencia SELECT.

Devuelven un solo tipo de valor definido en la cláusula "Returns". **CORRECTA**

Es similar a un procedimiento, salvo que puede referenciarse como si se tratara de una tabla.



Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Tipos de funciones - Funciones Tabulares en Linea:

Devuelve una tabla que es el resultado de una única sentencia SELECT. **CORRECTA**

Devuelven un solo tipo de valor definido en la cláusula "Returns".

Es similar a un procedimiento, salvo que puede referenciarse como si se tratara de una tabla.

Este tipo de funciones es sintácticamente similar a funciones del tipo COUNT(), MAX().



Más de una respuesta de las ANTERIORES es correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Práctica - Procedimientos y Funciones - CORRECTAS: 2 de 4

1) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.

```

CREATE PROCEDURE InsertOrUpdateAuto
    @Patente VARCHAR(10),
    @Modelo VARCHAR(50),
    @Conductor VARCHAR(100)
AS
BEGIN
    BEGIN TRY
        INSERT INTO Agencia (Patente, Modelo, Conductor)
        VALUES (@Patente, @Modelo, @Conductor);
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER() = 2627
        BEGIN
            UPDATE Agencia
            SET Conductor = @Conductor
            WHERE Patente = @Patente AND Conductor <> @Conductor;
        END
        ELSE
        BEGIN
            DECLARE @ErrorMessage NVARCHAR(4000);
            DECLARE @ErrorSeverity INT;
            DECLARE @ErrorState INT;
            SELECT
                @ErrorMessage = ERROR_MESSAGE(),
                @ErrorSeverity = ERROR_SEVERITY(),
                @ErrorState = ERROR_STATE();
            RAISERROR(@ErrorMessage, @ErrorSeverity, @ErrorState);
        END
    END CATCH
END;

```

Confeccionar un SP que permita insertar un nuevo auto en la tabla Agencia. Si ya existe un registro con la misma Patente, se debe eliminar ese registro y luego insertar el nuevo. Si se produce algún error durante la operación, se debe manejar mostrando un mensaje que indique que se ha eliminado el registro anterior.

Confeccionar un SP que permita insertar un nuevo auto en la tabla. Si ya existe un registro con la misma Patente, se debe insertar un nuevo registro sin actualizar ningún campo. Si se produce un error durante la operación, se debe manejar adecuadamente, mostrando un mensaje que indique que la operación se realizó correctamente.

Confeccionar un SP que permita insertar un nuevo auto en la tabla. Si ya existe un registro con la misma Patente, se debe insertar un nuevo registro sin actualizar ningún campo. Si se produce un error durante la



produce algún error durante la operación, se debe manejar. **CORRECTA**

Ninguno de los enunciados ANTERIORES es correcto. Código Incorrecto

2) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.

```

CREATE PROCEDURE InsertOrUpdateAuto
    @Patente VARCHAR(10),
    @Modelo VARCHAR(50),
    @Conductor VARCHAR(100)
AS
BEGIN
    BEGIN TRY
        INSERT INTO Agencia (Patente, Modelo, Conductor)
        VALUES (@Patente, @Modelo, @Conductor);
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER() = 2627
        BEGIN
            UPDATE Auto
            SET Conductor = @Conductor
            WHERE Patente = @Patente AND Conductor <> @Conductor;
        END
        ELSE
        BEGIN
            DECLARE @ErrorMessage NVARCHAR(4000);
            DECLARE @ErrorSeverity INT;
            DECLARE @ErrorState INT;
            SELECT
                @ErrorMessage = ERROR_MESSAGE(),
                @ErrorSeverity = ERROR_SEVERITY(),
                @ErrorState = ERROR_STATE();
            RAISERROR(@ErrorMessage, @ErrorSeverity, @ErrorState);
        END
    END CATCH
END;

```

Confeccionar un SP que permita insertar un nuevo auto en la tabla Agencia. Si ya existe un registro con la misma Patente, se debe eliminar ese registro y luego insertar el nuevo. Si se produce algún error durante la operación, se debe manejar mostrando un mensaje que indique que se ha eliminado el registro anterior.

X Confeccionar un SP que permita insertar un nuevo auto en la tabla. Si ya existe un registro con la misma Patente, se debe insertar un nuevo registro sin actualizar ningún campo. Si se produce un error durante la operación, se debe manejar adecuadamente, mostrando un mensaje que indique que la operación se realizó correctamente. **INCORRECTA**

Confeccionar un SP que permita insertar un nuevo auto en la tabla. Si ya existe un registro con la misma Patente, se debe insertar un nuevo registro sin actualizar ningún campo. Si se produce un error durante la operación, se debe manejar adecuadamente, mostrando un mensaje que indique que la operación se realizó incorrectamente.

Confeccionar un SP que permita insertar un nuevo Auto en la tabla Agencia. Si ya existe un registro con la misma Patente, se debe actualizar solo el campo Conductor si el valor ingresado es diferente al ya cargado. Si se produce algún error durante la operación, se debe manejar.

Ninguno de los enunciados ANTERIORES es correcto. Código Incorrecto **CORRECTA**

3) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.



```

IF EXISTS (SELECT 1 FROM dbo.Inventariocelulares WHERE Modelo = @Modelo)
BEGIN
    IF EXISTS (SELECT 1 FROM dbo.Inventariocelulares WHERE Modelo = @Modelo AND Marca <> @Marca)
    BEGIN
        THROW 50002, 'Error: El Modelo ingresado ya existe con una Marca diferente.', 1;
    END
    ELSE
    BEGIN
        UPDATE dbo.Inventario
        SET Memoria = @Memoria,
            Procesador = @Procesador
        WHERE Modelo = @Modelo;
    END
END
ELSE
BEGIN
    INSERT INTO dbo.Inventario (Marca, Modelo, Memoria, Procesador)
    VALUES (@Marca, @Modelo, @Memoria, @Procesador);
END
END;

```

Confeccionar un SP que permita insertar un nuevo dispositivo en la tabla dbo.Inventariodispositivos. Si ya existe un registro con el mismo Marca y Modelo, debe actualizar todos los campos (Memoria, Procesador y Marca). Si el Modelo ya está registrado pero la Marca es diferente, debe actualizar solo los valores de Memoria y Procesador sin generar ningnú error. En caso duplicados manejar el error.

Confeccionar un SP que permita insertar un nuevo celular en la tabla dbo.Inventariocelulares. Si ya existe cargado el mismo Modelo con Marca diferente, se debe permitir la inserción de un nuevo registro sin generar ningnú error. Si el Modelo y la Marca coinciden, se deben actualizar todos los campos del celular (Marca, Memoria, Procesador). En caso de registro, no se debe realizar ninguna acción.

Confeccionar un SP para insertar un nuevo Celular en la tabla dbo.Inventariocelulares. Si ya existe un registro con el mismo Modelo, se debe actualizar solo los campos nuevos del celular con los valores de memoria y procesador, y si el valor ingresado es una Marca y Modelo diferente a los ya cargados se debe cargar el nuevo registro, y si los datos son duplicados se generar el error y manejarlo.

- X Confeccionar un SP para insertar un nuevo celular en la tabla dbo.Inventariocelulares. Si ya existe un registro con el mismo Modelo, se debe actualizar solo el campo Marca y dejar intactos los valores de Memoria y Procesador. Si se intenta insertar un nuevo registro con un Modelo y Marca duplicados, el procedimiento debe permitir la inserción sin lanzar ningnú error. **INCORRECTA**

Ninguno de los enunciados ANTERIORES es correcto. Codigo Incorrecto **CORRECTA**

4) Dado el siguiente código de Store Procedure de SQL indicar el enunciado correcto.



```
AS
BEGIN
    IF @numero % 2 <> 0
    BEGIN
        THROW 50001, 'El número ingresado no es par. Inserción fallida.', 1;
    END
    ELSE
    BEGIN
        INSERT INTO dbo.NumeroPar (nro) VALUES (@numero);
    END
END;
EXEC InsertarNumeroPar @numero = 4;
```

Creacion de Store Procedure que inserte a la tabla dbo.NumeroPar el número ingresado automaticamente, pero sólo se deberá insertar si el número es par. De lo contrario lanzar una excepción. La tabla debe ser creada en el procedimiento.

Creacion de Store Procedure que inserte a la tabla dbo.NumeroPar el número ingresado por parámetro, pero sólo se deberá insertar si el número es par. De lo contrario lanzar una excepción. La tabla debe ser creada en el procedimiento.

Creacion de Store Procedure que inserte a la tabla dbo.NumeroPar el número ingresado automaticamente, pero sólo se deberá insertar si el número es par. De lo contrario lanzar una excepción. La tabla debe ser creada fuera del procedimiento.

Creacion de Store Procedure que inserte a la tabla dbo.NumeroPar el número ingresado por parámetro, pero sólo se deberá insertar si el número es par. De lo contrario lanzar una excepción. La tabla debe ser creada fuera del procedimiento. **CORRECTA**

Ninguno de los enunciados ANTERIORES es correcto. Codigo Incorrecto

2P - Practica - Triggers - CORRECTAS: 1 de 1

1) ¿Cuál es el resultado de la siguiente instrucción SQL para crear un trigger en la tabla Facturas?



```

IF EXISTS (SELECT * FROM deleted)
BEGIN
    SELECT @FacturaID = FacturaID FROM deleted;
    SET @TipoAccion = 'Eliminación';

    INSERT INTO Factura_Log (FacturaID, Accion, FechaAccion)
    VALUES (@FacturaID, @TipoAccion, GETDATE());
END

IF EXISTS (SELECT * FROM inserted)
BEGIN
    SELECT @FacturaID = FacturaID FROM inserted;
    SET @TipoAccion = 'Actualización';

    INSERT INTO Factura_Log (FacturaID, Accion, FechaAccion)
    VALUES (@FacturaID, @TipoAccion, GETDATE());
END
END;

```

El trigger registra la inserción de nuevas facturas en la tabla Factura_Log con la acción correspondiente.

El trigger registra las eliminaciones de facturas en la tabla Factura_Log con la acción correspondiente.

- El trigger registra tanto las eliminaciones como las actualizaciones de facturas en la tabla Factura_Log con la acción correspondiente. **CORRECTA**

El trigger impide que se eliminen o actualicen facturas en la tabla Facturas.

Ninguno de los enunciados ANTERIORES es correcto.

2P - Seguridad - General SQLServer - CORRECTAS: 0 de 1

1) Entidades de seguridad y seguridad de objetos de base de datos:

El ámbito de influencia de una entidad de seguridad no depende del ámbito de su definición: Windows, servidor o base de datos; y de si la entidad de seguridad es indivisible o es una colección.

Un Inicio de sesión de Windows es un ejemplo de entidad de seguridad divisible y un Grupo de Windows es un ejemplo de una del tipo colección.

Las entidades de seguridad son entidades que pueden solicitar recursos de SQL Server. **CORRECTA**

Este tema se aplica a todas las versiones de SQL Server, y no hay algunas restricciones en las entidades de seguridad a nivel de servidor de SQL Database.

- Mas de una respuesta de las ANTERIORES es Correcta. **INCORRECTA**

Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Seguridad - Usuarios y Esquemas - CORRECTAS: 1 de 2

1) Inicio de sesión sa:

La cuenta sa se crea de forma predeterminada cuando se instala una base de datos.

La base de datos predeterminada de la cuenta sa es msdb.



Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta. **CORRECTA**

2) Inicio de sesión sa:

La cuenta sa se crea de forma predeterminada cuando se instala una base de datos.

La base de datos predeterminada de la cuenta sa es msdb.

El inicio de sesión, la cuenta sa es miembro del rol fijo de nivel de servidor sysdba.

El inicio de sesión de SQL Server, la cuenta sa es una entidad de seguridad a nivel de servidor. **CORRECTA**

Mas de una respuesta de las ANTERIORES es Correcta.

Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Seguridad - ROLES - CORRECTAS: 0 de 1

1) Rol público de base de datos y de servidor:

Cada inicio de sesión pertenece al rol fijo de servidor public y cada usuario de base de datos pertenece al rol de base de datos master.

Cuando a un usuario o inicio de sesión se le han concedido permisos concretos para un elemento protegible, hereda los permisos para ese elemento concedidos a public.

El rol fijo de servidor public y el de base de datos public pueden quitarse.

Hay muchos de los permisos que se asignan a los roles public de forma predeterminada. La mayoría de estos permisos no son necesarios para realizar operaciones rutinarias en la base de datos; el tipo de tareas que los dba deben poder hacer.

Mas de una respuesta de las ANTERIORES es Correcta.

Todas las respuestas ANTERIORES son correcta. **INCORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta. **CORRECTA**

2P - Seguridad - Elementos protegibles - CORRECTAS: 0 de 1

1) Elementos protegibles => Ámbito protegible: base de datos

Grupo de disponibilidad, Punto de conexión, Iniciar sesión, Rol del servidor, Base de datos.

Rol de aplicación, Ensamblado, Clave asimétrica, Iniciar sesión, Contrato.

Ruta, Esquema, Lista de propiedades de búsqueda, Service, Rol del servidor.

Catálogo de texto completo, Lista de palabras irrelevantes de texto completo, Tipo de mensaje, Enlace de servicio remoto, (Base de datos) Rol. **CORRECTA**



2P - Seguridad - Jerarquía de Cifrado - CORRECTAS: 0 de 1

1) Jerarquía de cifrado => Conceptos:

- Para obtener el máximo rendimiento, cifre los datos utilizando claves asimétricas en lugar de certificados o claves simétricas. **INCORRECTA**

Las claves maestras de base de datos se protegen mediante la clave asimétrica maestra de servicio.

El programa de instalación de SQL Server crea la clave maestra de servicio, que se cifra con la API de protección de datos de Windows (DPAPI). **CORRECTA**

El módulo de Administración extensible de claves (EKM) mantiene las claves asimétricas o simétricas fuera de SQL Server.

Mas de una respuesta de las ANTERIORES es Correcta.

Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Teoria - Triggers - CORRECTAS: 0 de 2

1) Triggers: AFTER

Pueden usarse como mecanismo para realizar operaciones no controladas en tablas.

La creacion inadecuada pueden causar daño a la base de datos y que la misma se comporte de manera inadecuadamente.

Se disparan en paralelo de la operación que los produce.

Sirven en general para implementar restricciones de negocio avanzadas. **CORRECTA**

Mas de una respuesta de las ANTERIORES es correcta.

- Todas las respuestas ANTERIORES son correctas. **INCORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Triggers:

Los TRIGGERS son muy similares en su concepto a los procedimientos almacenados, son código Transact-SQL similares en código y ejecución.

Un TRIGGER se ejecuta en respuesta a un evento.

La manipulación de datos son del tipo TRIGGERS DMZ o por la manipulación de esquemas como los TRIGGERS DLL.

Reciben parámetros de una forma muy especial, en forma de unas tablas virtuales, llamadas inserted y deleted.



2P - Transacciones y Bloqueos - CORRECTAS: 0 de 2

1) Características ACID: => Atomicidad (Atomicity)

Asegura que las transacciones concurrentes se ejecuten de manera que no afecten entre sí.

Garantiza que los cambios realizados por la transacción se guardan de forma que no se perderán, garantizando la integridad de los datos.

Esta propiedad asegura que una transacción se ejecute de manera completa o no se ejecute en absoluto.

CORRECTA

Garantiza que una transacción lleve a la base de datos de un estado válido a otro estado válido.

Mas de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas. **INCORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2) Transacciones Implícitas:

Inicia manualmente una nueva transacción después de confirmar o revertir la transacción anterior.

Se deben realizar las acciones necesarias para delinear el inicio de una transacción.

Debe validar la totalidad de la ejecución para que confirmar o revertir cada transacción.

Genera una cadena fraccionada de transacciones. **INCORRECTA**

Mas de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta. **CORRECTA**

2P - Teoria - Cursos - CORRECTAS: 0 de 1

1) Características de los cursos:

Nos permite movernos por los registros de una consulta como un único bloque para analizar el resultado todo junto.

Nos permite recorrer los registros del resultado obtenido de una sentencia WHERE.

Se usa para realizar una acción si el código identificador de un registro no es autonumérico.

Su uso permiten procesar el conjunto de resultados de fila en fila. **CORRECTA**

Mas de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas. **INCORRECTA**

Ninguna de TODAS las respuestas ANTERIORES es correcta.



```

DECLARE @ProductName nvarchar(100)
DECLARE ProductCursor CURSOR FOR
SELECT ProductName FROM Products
OPEN ProductCursor
FETCH NEXT FROM ProductCursor INTO @ProductName
WHILE @@FETCH_STATUS = 0
BEGIN
    PRINT @ProductName
    FETCH NEXT FROM ProductCursor INTO @ProductName
END
CLOSE ProductCursor
DEALLOCATE ProductCursor

```

Insertar todos los nombres de productos en una nueva tabla.

- Imprimir todos los nombres de productos de la tabla Products uno por uno. **CORRECTA**
- Eliminar todos los registros en la tabla Products.
- Actualizar el nombre del producto en la tabla Products.
- Ninguna de las opciones ANTERIORES es correcta.

RESUMEN

TOTAL CORRECTAS: 8 de 30 **INSUFICIENTE**

2P - Variables SQL	CORRECTAS: 0 de 2
2P - Variables SQL - Características	CORRECTAS: 0 de 2
2P - Procedimientos Almacenados	CORRECTAS: 0 de 2
2P - Procedimientos Almacenados - Características	CORRECTAS: 1 de 2
2P - Procedimientos Almacenados - Ventajas	CORRECTAS: 0 de 1
2P - Procedimientos Almacenados - Parámetros	CORRECTAS: 2 de 2
2P - Tipos de funciones	CORRECTAS: 0 de 2
2P - Práctica - Procedimientos y Funciones	CORRECTAS: 2 de 4
2P - Práctica - Triggers	CORRECTAS: 1 de 1
2P - Seguridad - General SQLServer	CORRECTAS: 0 de 1
2P - Seguridad - Usuarios y Esquemas	CORRECTAS: 1 de 2
2P - Seguridad - ROLES	CORRECTAS: 0 de 1
2P - Seguridad - Elementos protegibles	CORRECTAS: 0 de 1
2P - Seguridad - Jerarquía de Cifrado	CORRECTAS: 0 de 1
2P - Teoría - Triggers	CORRECTAS: 0 de 2
2P - Transacciones y Bloqueos	CORRECTAS: 0 de 2
2P - Teoría - Cursos	CORRECTAS: 0 de 1
2P - Práctica - Cursos	CORRECTAS: 1 de 1



JUAREZ, ELENA MABEL

[Mi perfil](#) · [Ayuda](#) · [Cerrar sesión](#)

UNLaM - DIIT - UEaD. Miel Materias Interactivas en Línea - 2024 - [Aviso legal](#)