



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.

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



Más 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) 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**

Mas 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.

☒ No existen variables globales en SQL Server. **CORRECTA**

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).

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 ejecución. **CORRECTA**

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.

☒ 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:

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. **INCORRECTA**



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:

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 de puede llamar a otros SP).

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:

No aceptan parámetros de entrada.

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. **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.

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 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.

Mas 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: 0 de 2

1) Tipos de funciones - Funciones Tabulares MultiSentencias:

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

- ✗ Devuelve una tabla construida por una o mas sentencias Transact-SQL. **INCORRECTA**

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

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

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) Tipos de funciones - Funciones Tabulares MultiSentencias:

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

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



Todas las respuestas ANTERIORES son correctas.



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

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

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

```
TABLAS:
CultureID (tipo: int)
CultureName (tipo: nvarchar)
ModifiedDate (tipo: datetime)

CREATE PROCEDURE p_InsCulture
    @id INT,
    @name NVARCHAR(100),
    @date DATETIME
AS
BEGIN
    INSERT INTO Production.Culture (CultureID, CultureName, ModifiedDate)
    VALUES (@id, @name, @date);
END;
EXEC p_InsCulture @id = 1, @name = 'Wheat', @date = GETDATE();
```

Confeccionar el Store procedure p_InsCulture(id,name,date) que permita actualiza el CultureName y la ModifiedDate de un registro en la tabla Production.Culture, basándose en el CultureID dado.

Confeccionar el Store procedure p_InsCulture(id,name,date) que permita agrega un nuevo campo a la tabla Production.Culture y asigna los valores de CultureID, CultureName y ModifiedDate como parámetros de entrada.

Confeccionar el Store procedure p_InsCulture(id,name,date) que permita dar de alta un nuevo registro en la tabla Production.Culture. Los tipos de datos de los parámetros deben corresponderse con la tabla.

CORRECTA

Confeccionar el Store procedure p_InsCulture(id,name,date) que permita inserta un nuevo registro en la tabla Production.Culture únicamente si el CultureName no existe previamente en la tabla.

Confeccionar el Store procedure p_InsCulture(id,name,date) que devuelva el número total de registros en la tabla Production.Culture después de insertar un nuevo CultureID, CultureName y ModifiedDate.



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

2) 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**

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

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

```

CREATE PROCEDURE dbo.InsertarOActualizarAlumno
    @DNI INT,
    @NombreApellido NVARCHAR(100),
    @Celular NVARCHAR(15),
    @Email NVARCHAR(100)
AS
BEGIN
    SET NOCOUNT ON;
    IF EXISTS (SELECT 1 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.estudiante
            SET Celular = @Celular,
                Email = @Email
            WHERE DNI = @DNI;
        END
    END
    ELSE
    BEGIN
        INSERT INTO dbo.estudiante (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.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. **INCORRECTA**

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 InsertarActualizarPelicula
    @Titulo VARCHAR(255),
    @Sala VARCHAR(50),
    @Genero VARCHAR(50)
AS
BEGIN
    BEGIN TRY
        INSERT INTO dbo.Pelicula (Titulo, Sala, Genero)
        VALUES (@Titulo, @Sala, @Genero);
    END TRY
    BEGIN CATCH
        IF ERROR_NUMBER() = 2627
        BEGIN
            UPDATE dbo.Cine
            SET Sala = @Sala
            WHERE Titulo = @Titulo AND Sala <> @Sala;
        END
        ELSE
        BEGIN
            -- Manejar otros errores
            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 una nueva Pelicula en la tabla `dbo.Cine`. Si ya existe un registro con la mismo Titulo, se debe agregar un nuevo registro con todos los datos. Si se produce algún error durante la operación, se debe manejar.

Confeccionar un SP que permita insertar una nueva Pelicula en la tabla `dbo.Cine`. Si ya existe un registro con la misma Sala, se debe actualizar solo el campo Sala si el valor ingresado es diferente al ya cargado. Si se produce algún error durante la operación, se debe manejar.

✗ Confeccionar un SP que permita insertar una nueva Pelicula en la tabla `dbo.Pelicula`. Si ya existe un registro con la mismo Titulo, se debe actualizar solo el campo Sala si el valor ingresado es diferente al ya cargado. Si se produce algún error durante la operación, se debe manejar. **INCORRECTA**

Confeccionar un SP que permita insertar una nueva Pelicula en la tabla `dbo.Cine`. Si ya existe un registro con la mismo Titulo, se debe actualizar solo el campo Sala 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**

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

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



AS

BEGIN

INSERT INTO Pedido_Log (PedidoID, Accion, Fecha)

SELECT PedidoID, 'Nuevo Pedido', GETDATE()

FROM inserted;

END;

El trigger evita que se inserten nuevos registros en la tabla Pedidos.

El trigger registra solo las actualizaciones realizadas en la tabla Pedidos, indicando el ID del pedido, la acción "Nuevo Pedido", y la fecha de la inserción.

El trigger actualiza la tabla Pedidos con nuevos registros a medida que se insertan, indicando el ID del pedido, la acción "Nuevo Pedido", y la fecha de la inserción.



El trigger registra los nuevos pedidos insertados en la tabla Pedido_Log, indicando el ID del pedido, la acción "Nuevo Pedido", y la fecha de la inserción. **CORRECTA**

Ninguno de los enunciados ANTERIORES es correcto.

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

1) Para el acceso a una base de datos Datos SQL Server, indicar los pasos correctos:

Crear alguno de los dos tipos de inicio de sesión: Inicio de Sesión Windows (usuarios o grupos) o Inicios de Sesión SQL Server (que solo podrán crearse en Autenticación Mixta).

La cadena de conexión especifica una autenticación a nivel de Instancia de SQL Server Express.

Una vez se ha autenticado el inicio de sesión, se comprueba si ese inicio de sesión tiene acceso a la base de datos a la que se pretende acceder.

SQL Server mantiene una asociación entre Inicio de Sesión a nivel de Instancia y Usuario a nivel de Base de Datos.

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 - Usuarios y Esquemas - CORRECTAS: 0 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.



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


INCORRECTA



2) 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. **INCORRECTA**

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.

Todas las respuestas ANTERIORES son correcta. **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 no se le han concedido ni denegado permisos concretos para un elemento protegible, hereda los permisos para ese elemento concedidos a public. **CORRECTA**

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. **INCORRECTA**

Todas las respuestas ANTERIORES son correcta.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

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

1) Elementos protegibles => Ámbito protegible: servidor.

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

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

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. **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 - Seguridad - Jerarquía de Cifrado - CORRECTAS: 0 de 1



Para obtener el máximo rendimiento, cifre los datos utilizando claves simétricas en lugar de certificados o claves asimétricas. **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 disponibilidad de los datos dentro de nuestras tablas de usuario.

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

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

A esta categoría de eventos pertenecen entre otros los de CREATE LOGIN o CREATE/ALTER/DROP Database.

✗ 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: 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 actualizacion 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 instalacion 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.

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

Ninguna de TODAS las respuestas ANTERIORES es correcta.

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

1) Transacciones Explícitas:

Es aquella en que se define Precisamente el inicio y el final de la transacción.



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) Transacciones Explícitas:

Es aquella en 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 automatico de ejecucion para cumplir con la propiedad ACID.

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 - Cursores - CORRECTAS: 1 de 1

1) Características de los cursores:

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.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

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

1) ¿Qué valor tiene la variable de sistema @@FETCH_STATUS cuando el FETCH en un cursor es exitoso?



0 **CORRECTA**

-1

1

NULL

Ninguna de las opciones ANTERIORES es correcta.

RESUMEN



UNLaM



MleL

RAMOS, DIEGO ENRIQUE

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

2P - Procedimientos Almacenados - Caracteristicas	CORRECTAS: 2 de 2
2P - Procedimientos Almacenados - Ventajas	CORRECTAS: 0 de 1
2P - Procedimientos Almacenados - Parametros	CORRECTAS: 2 de 2
2P - Tipos de funciones	CORRECTAS: 0 de 2
2P - Practica - Procedimientos y Funciones	CORRECTAS: 1 de 4
2P - Practica - 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: 0 de 1
2P - Seguridad - Elementos protegibles	CORRECTAS: 0 de 1
2P - Seguridad - Jerarquía de Cifrado	CORRECTAS: 0 de 1
2P - Teoria - Triggers	CORRECTAS: 0 de 2
2P - Transacciones y Bloqueos	CORRECTAS: 2 de 2
2P - Teoria - Cursores	CORRECTAS: 1 de 1
2P - Practica - Cursores	CORRECTAS: 1 de 1

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