

## 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
    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 ningún 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.

Mas 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](#)