



## 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.

✘ Ninguna de TODAS las respuestas ANTERIORES es correcta.

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

### 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.

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).

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: 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 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. **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.

Son sentencias de control de flujo de ejecución, para la implementación de procesos repetitivos.

s o

Son módulos o rutinas que encapsulan código para su reutilización.

Más de una respuesta de las ANTERIORES es correcta.



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 más 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.

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 - 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.

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 - 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.



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

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 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 - 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.

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 sintacticamente 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".

Es similar a una vista ofreciendo mas flexibilidad.

Se le pueden suministrar parametros a la funcion.

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 - 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
    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.

```

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.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;

```

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.

PIEDRABUENA, WALTER NICOLAS

Mi perfil Ayuda Cerrar sesión

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

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



Confeccionar la creacion de tabla dbo.Cientes y el Store procedure que permita insertar Personas (con el campo DNI, Apellido y Nombre) en una tabla con el nombre dbo.Cientes, 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.Cientes y el Store procedure que permita insertar Personas (con el campo DNI, Apellido y Nombre) en una tabla con el nombre dbo.Ciente, 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?

```
CREATE TRIGGER trg_OrderInsert
ON Orders
AFTER INSERT
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 esta 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.



Ninguna de TODAS las respuestas ANTERIORES es correcta.

## 2) Usuario y esquema dbo:

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.

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.

## 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 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**

Más de una respuesta de las ANTERIORES es correcta.

Todas las respuestas ANTERIORES son correctas.

Ninguna de TODAS las respuestas ANTERIORES es correcta.

## 2P - Teoría - 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 DML o por la manipulación de esquemas como los TRIGGERS DDL.

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

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) 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.

✗ 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 - Transacciones y Bloqueos - CORRECTAS: 1 de 2

---

1) Transacciones Implícitas:

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

No tiene que realizar ninguna acción para delinear el inicio de una transacción. **CORRECTA**

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

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 - Cursores - CORRECTAS: 0 de 1

### 1) Características de los cursores - 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 - Cursores - 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.