

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.

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 antes de usarse.

Más de una respuesta de las ANTERIORES es correcta.

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

☒ 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 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**

2P - Variables SQL - Características - CORRECTAS: 1 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 "0" (Cero).



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) 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.

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



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 - 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 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.



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:

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 bloques condicionales o repetitivos.

Son módulos o rutinas que encapsulan código para su reutilización.

✗ Mas de una respuesta de las ANTERIORES es correcta. **INCORREC**

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

Ninguna de TODAS las respuestas ANTERIORES es correcta.

2P - Procedimientos Almacenados - Caracteristicas - CORRECTAS: 0 de 2

1) Procedimientos Almacenados - Caracteristicas:

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). **CORRECTA**

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. **INCORRECTA**

2) Procedimientos Almacenados - Caracteristicas:

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. **INCORRECTA**

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 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.

✓ 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 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. **INCORRECTA**

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.

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 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. **CORRECTA**

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

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

Mas de una respuesta de las ANTERIORES es 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".

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

CORRECTA

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

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 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 un 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 un 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 un 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.

Confeccionar un SP que permita insertar un 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**

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 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. **INCORRECTA**

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 InsertarActualizarPelicula
    @Titulo VARCHAR(255),
    @Sala VARCHAR(50),
    @Genero VARCHAR(50)
AS
BEGIN
    BEGIN TRY
        INSERT INTO dbo.Cine (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 un 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. **INCORRECTA**

Confeccionar un SP que permita insertar un 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 un 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. **CORRECTA**

Ninguno de los enunciados ANTERIORES es correcto. Código Incorrecto

4) 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.

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 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. **INCORRECTA**

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 Ventas?

```

CREATE TRIGGER trg_NuevaVenta
ON Ventas
AFTER INSERT
AS
BEGIN
    DECLARE @MontoTotal DECIMAL(10,2);

    SELECT @MontoTotal = SUM(Monto)
    FROM inserted;

    IF @MontoTotal > 10000
    BEGIN
        PRINT 'Alerta: Se ha realizado una venta mayor a 10,000!';
    END;
END;

```

El trigger impide que se inserten registros con un monto superior a 10,000.

El trigger calcula el monto total y actualiza los registros en la tabla Ventas.

✓ El trigger ejecuta la acción después de que se inserta un nuevo registro, y si el monto total de la venta es superior a 10,000, muestra un mensaje de alerta. **CORRECTA**

El trigger muestra el monto total de la venta, independientemente de su valor.

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 Active Directory para el servidor Windows que no está en el dominio donde está instalado el motor de Base de Datos.

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 por medio de la base de datos master.

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

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

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

2) Usuario y esquema dbo:

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

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.

CORRECTA

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. **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:

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. **INCORRECTA**

Todas las respuestas ANTERIORES son correcta.

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, Certificate, Contrato.

CORRECTA

Ruta, Esquema, Iniciar sesión, Service, Clave simétrica.

Catálogo de texto completo, Lista de palabras irrelevantes de texto completo, Tipo de mensaje, Enlace de servicio remoto, Base de datos.

Mas de una respuesta de las ANTERIORES es Correcta.



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

Ninguna de TODAS las respuestas ANTERIORES es 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.

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.

El módulo de Administración extensible de claves (EKM) mantiene las claves simétricas o asimétricas fuera de SQL Server. **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 - Teoria - Triggers - CORRECTAS: 1 de 2

1) ¿Cuál de las siguientes afirmaciones describe mejor un trigger en SQL Server?



Un trigger es una instrucción SQL que se ejecuta de manera automática en respuesta a ciertos eventos en una tabla o vista. **CORRECTA**

Un trigger es un procedimiento almacenado que se ejecuta manualmente cuando se invoca.

Un trigger es una vista que se actualiza automáticamente cuando cambia la tabla subyacente.

Un trigger es una función que se llama cada vez que se realiza una consulta en una tabla.

Ninguno de los enunciados ANTERIORES es correcto.

2) Triggers:

Un TRIGGER se ejecuta en respuesta a un evento por la petición de un cliente o del motor de Base de datos.

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

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 funciones virtuales, llamadas inserted y deleted.



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 Explícitas:

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

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

Reciben el nombre de transacciones definidas por el usuario. **CORRECTA**

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

Se inician y finalizan Precisamente mediante comandos como BEGIN TRANSACTION, ROLLBACK, END.

 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: => Durabilidad (Durability)

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.

 Garantiza que los cambios realizados por la transacción se guardan de forma que no se perderán, garantizando la integridad de los datos. **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 - Teoria - Cursores - CORRECTAS: 0 de 1

1) Características de los cursores:

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

Nos permite recorrer los registros del resultado obtenido de una sentencia SELECT. **CORRECTA**

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 un sentencia tabla por tabla.

 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 - Cursores - CORRECTAS: 1 de 1

1) ¿Qué hace el siguiente código de cursor SQL?

```
DECLARE @IDcliente INT;  
DECLARE ClientesCursor CURSOR FOR SELECT ClienteID FROM Clientes WHERE Activo = 1;  
OPEN ClientesCursor;  
FETCH NEXT FROM ClientesCursor INTO @IDcliente;  
  
WHILE @@FETCH_STATUS = 0  
BEGIN  
    DELETE FROM Clientes WHERE ClienteID = @IDcliente;  
    FETCH NEXT FROM ClientesCursor INTO @IDcliente;  
END;  
  
CLOSE ClientesCursor;  
DEALLOCATE ClientesCursor;
```

Inserta un nuevo cliente en la tabla Clientes con el ClienteID especificado.

Selecciona todos los clientes de la tabla Clientes y los muestra en pantalla.



Elimina todos los clientes activos en la tabla Clientes.

CORRECTA

Actualiza el ClienteID en la tabla Clientes para los registros activos.

Ninguna de las opciones ANTERIORES es correcta.



UNLAM



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

