

***Examen Final***

---

**Administración de Base de Datos**

Página | 1

Apellidos y Nombres: \_\_\_\_\_ Código: \_\_\_\_\_

Ciclo: \_\_\_\_\_ Salón: \_\_\_\_\_

**Enunciado 01:**

De acuerdo con la **base de datos** desarrollada en **Microsoft SQL Server**, responda las siguientes preguntas:

- 1) Explique que problema soluciona su base de datos  
Mi base de datos trata de los recursos humanos de la empres Delosi:  
Primero hago una interfaz de una asistencia personal de esa manera puedo mejorar el registro del personal y por otra tengo una conexión con otra tabla que trata de el inventario de PIZZA HUT para evitar desabastecimientos o excesos de productos.  
Cuenta con una gestor de venta lo que se hace día a día.
- 2) Implemente un Script para crear una **vista** para crear utilizando tres tablas

```
CREATE VIEW vista_general AS
SELECT
    a.id_almacen,
    a.nombre AS nombre_almacen,
    d.nombre_empleado,
    v.id_venta,
    v.total_venta
FROM
    almacen a
JOIN
    delositable d ON a.id_empleado = d.id_empleado
JOIN
    ventas v ON d.id_empleado = v.id_empleado;
```

- 3) Implemente un Script para crear un **procedimiento almacenado** para modificar el ingreso de datos en forma secuencial

```

DELIMITER //
CREATE PROCEDURE actualizar_inventario(
    IN p_id_almacen INT,
    IN p_nuevo_stock INT
)
BEGIN
    UPDATE almacen
    SET stock = p_nuevo_stock
    WHERE id_almacen = p_id_almacen;
END //
DELIMITER ;

```

- 4) Implemente un Script para crear un **disparador** para verificar el control de datos (Ejemplo: que la nota ingresada este entre 0 y 20)

```

DELIMITER //
CREATE TRIGGER verificar_total_venta
BEFORE INSERT ON ventas
FOR EACH ROW
BEGIN
    IF NEW.total_venta < 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'El total de la venta no puede ser negativo';
    END IF;
END //
DELIMITER ;

```

- 5) Utilizando Script Crear 03 usuarios con nombres de sus compañeros y uno suyo

```

CREATE USER 'usuario1'@'localhost' IDENTIFIED BY 'admin123';
CREATE USER 'usuario2'@'localhost' IDENTIFIED BY 'admin1234';
CREATE USER 'jaime'@'localhost' IDENTIFIED BY 'admin123';

```

- 6) Utilizando un script, copiar la base de datos (creada anteriormente) y compartir en cada uno de los usuarios

```

CREATE DATABASE copia_basedatos;
USE copia_basedatos;

```

- 7) Utilizando un script, generar una copia de seguridad de la base de datos y compartir a cada uno de los usuarios

```

mysqldump -u root -p recursosh> copia_seguridad.sql

```

- 8) Utilizando un script, encriptar una de las tablas para que no se puedan ver los datos

```

ALTER TABLE ventas ENGINE=InnoDB;
ALTER TABLE ventas ENCRYPTION='Y';

```

*Mg. Ing. Raúl Fernández Bejarano*

- 9) Utilizando un script, aplique la seguridad a nivel de columna, restringiendo el acceso a la columna DNI de la tabla empleado en el usuario con nombre de su compañero

```
REVOKE SELECT (id) ON delositable FROM '1'@'mario';
```

- 10) Utilizando un script, implementé seguridad a nivel de columna restringiendo el acceso a una de las columnas de una tabla.

```
REVOKE SELECT (agregar) ON ventas FROM 'usuario2'@'localhost';
```

- 11) Utilizando un script, realice el cifrado transparente de datos (TDE) para una las tablas.

```
-- Configurar TDE
ALTER TABLE delositable ENCRYPTION='Y';
```

- 12) Utilizando un script, configure el usuario con el nombre de su compañero para otorgar permisos de SELECT, INSERT, UPDATE y DELETE en la base de datos.

```
GRANT SELECT, INSERT, UPDATE, DELETE ON recursosh* TO 'mario'@'p';
```

- 13) Utilizando un script, configure la auditoría para el seguimiento y registro de acciones en la base de datos

```
-- Activar el log binario
SET GLOBAL log_bin = 'ON';
-- Configurar auditoría
CREATE TABLE auditoria (
    id INT AUTO_INCREMENT PRIMARY KEY,
    accion VARCHAR(255),
    fecha TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```

- 14) Utilizando un script, configure de la memoria y el disco duro

```
SET GLOBAL innodb_buffer_pool_size = 512M;
SET GLOBAL innodb_log_file_size = 128M;
```

- 15) Utilizando un script, genere una copia de seguridad de la base de datos

```
mysqldump -u root -p recursosh > respaldo.sql
```

- 16) Utilizando un script, genere la restauración de la base de datos

```
mysql -u root -p recursosh< respaldo.sql
```

- 17) Utilizando un script, cree un espejo de la base de datos

```
-- Configuración de replicación maestra-esclava  
CHANGE MASTER TO  
    MASTER_HOST='1',  
    MASTER_USER='mario',  
    MASTER_PASSWORD='15';  
START SLAVE;
```

Página | 4

- 18) Utilizando un script, realice la replicación de bases de datos

```
-- Configuración de replicación en cascada  
CHANGE MASTER TO  
    MASTER_HOST='localhost3306',  
    MASTER_USER='recurosh',  
    MASTER_PASSWORD='';  
START SLAVE;
```

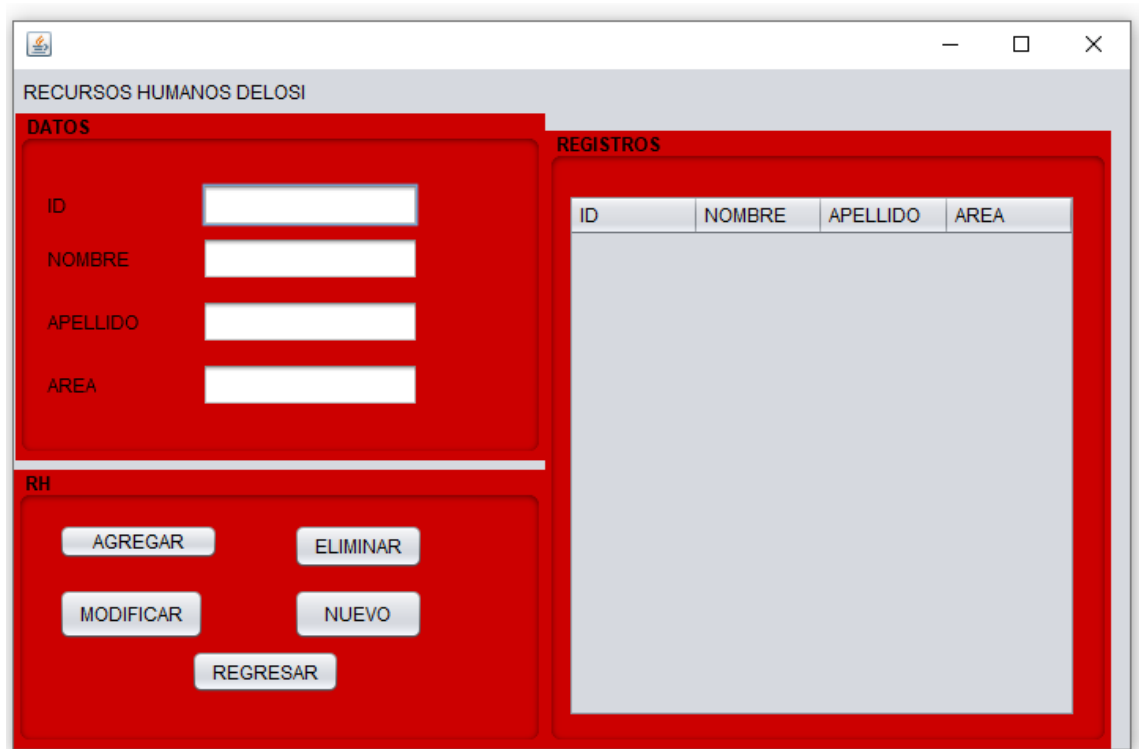
- 19) Explique que es Always On Availability Groups

Always On Availability Groups es una tecnología avanzada de SQL Server para alta disponibilidad y recuperación ante desastres. Permite sincronizar múltiples bases de datos dentro de un grupo y distribuir la carga de lectura.

- 20) Explique que es Log Shipping

Es una estrategia que utiliza copias periódicas del registro de transacciones para mantener actualizada una base de datos secundaria, asegurando continuidad en caso de fallos.

*Mg. Ing. Raúl Fernández Bejarano*



Din

ALMACEN DE LOSI

DATOS

CATEGORIA

PRODUCTO

CANTIDAD

RH

AGREGAR

ELIMINAR

MODIFICAR

NUEVO

REGRESAR

REGISTROS

CATEGORIA	PRODUCTO	CANTIDAD
-----------	----------	----------

PIZZA HUT - VENTAS

DATOS

N° PEDDI...	NOMBRE	PEDIDO	CANTIDAD
-------------	--------	--------	----------

VENTAS

N° PEDIDO:

NOMBRE:

PEDIDO:

CANTIDAD:

PIZZA HUT

AGREGAR ORDEN

ELIMINAR ORDEN

LIMPIAR

SALIR

Mg. Ing. Raúl Fernández Bejarano