



Universidad Peruana Los Andes
Facultad de Ingeniería
Escuela Profesional de Ingeniería de Sistemas y Computación

Administración de Base de Datos

Enunciado 01:

De acuerdo con la **base de datos** implementada (mínimo 100 registros), utilice los DBMS **Microsoft SQL Server/MySQL**, o un servidor de la nube como Microsoft Azure o Google FireBase. Explique qué problema soluciona su base de datos y responda las siguientes preguntas:

- 1) Implemente y explique un Script para crear una **vista** para crear utilizando tres tablas

```
Use EduTec
go

select * from CursoProgramado;

SELECT
    cp.IdCiclo,
    c.NomCurso AS Curso,
    p.IdProfesor AS Profesor,
    p.ApeProfesor AS ApellidoProfesor,
    p.NomProfesor AS NombreProfesor,
    e.IdEmpleado AS Empleado,
    e.NomEmpleado AS NombreEmpleado,
    e.Cargo AS CargoEmpleado,
    cp.Horario,
    cp.Vacantes,
    cp.Matriculados
FROM
    CursoProgramado cp
INNER JOIN
    Curso c ON cp.IdCurso = c.IdCurso
INNER JOIN
    Profesor p ON cp.IdProfesor = p.IdProfesor -- Relacionamos CursoProgramado con Profesor
LEFT JOIN
    Empleado e ON cp.IdEmpleado = e.IdEmpleado
```

| IdCursoProg | IdCurso | IdCiclo | IdProfesor | Vacantes | Matriculados | PreCursoProg | Horario | Activo |
|-------------|---------|---------|------------|----------|--------------|--------------|---------------------|--------|
| 1 | C005 | 1985-01 | P013 | 11 | 9 | 500.00 | Vi y Sa 19-22 | 1 |
| 2 | C014 | 1985-01 | P025 | 1 | 19 | 700.00 | Ju 19-22 y Sa 08-11 | 1 |
| 3 | C003 | 1985-01 | P022 | 5 | 15 | 700.00 | Lu y Mi 19-22 | 1 |
| 4 | C002 | 1985-01 | P017 | 12 | 8 | 500.00 | Ju 19-22 y Sa 08-11 | 1 |
| 5 | C008 | 1985-01 | P035 | 5 | 15 | 700.00 | Ju 19-22 y Sa 08-11 | 1 |
| 6 | C011 | 1985-01 | P016 | 6 | 14 | 700.00 | Lu y Mi 19-22 | 1 |
| 7 | C003 | 1985-01 | P022 | 5 | 15 | 700.00 | Ju 19-22 y Sa 08-11 | 1 |
| 8 | C003 | 1985-01 | P005 | 4 | 16 | 700.00 | Ju 19-22 y Sa 08-11 | 1 |
| 9 | C010 | 1985-01 | P042 | 3 | 17 | 700.00 | Ma y Ju 19-22 | 1 |

| IdCiclo | Curso | Profesor | ApellidoProfesor | NombreProfesor | Empleado | NombreEmpleado | CargoEmpleado | Horario | Vacantes | Matriculados |
|---------|--------------------------------------|----------|----------------------|----------------|----------|----------------|---------------|---------------------|----------|--------------|
| 1985-01 | Visual Basic .NET - Fundamentos | P017 | Rodriguez Villacorta | Manuel | NULL | NULL | NULL | Ju 19-22 y Sa 08-11 | 12 | 8 |
| 1985-01 | Java - Frameworks | P035 | Suarez Valenzuela | Misael | NULL | NULL | NULL | Ju 19-22 y Sa 08-11 | 5 | 15 |
| 1985-01 | PHP - Frameworks | P016 | Rosas Ayala | Dario | NULL | NULL | NULL | Lu y Mi 19-22 | 6 | 14 |
| 1985-01 | Visual Basic .NET - Cliente/Servidor | P022 | Lostanau Navarro | Alberto | NULL | NULL | NULL | Ju 19-22 y Sa 08-11 | 5 | 15 |
| 1985-01 | Visual Basic .NET - Cliente/Servidor | P005 | Bustamante Gutie... | Cesar Augusto | NULL | NULL | NULL | Ju 19-22 y Sa 08-11 | 4 | 16 |
| 1985-01 | PHP - POO | P042 | Veliz | Fortunato | NULL | NULL | NULL | Ma y Ju 19-22 | 3 | 17 |
| 1985-01 | SO Windows | P036 | Carrasco Muñoz | Joel | NULL | NULL | NULL | Ju 19-22 y Sa 08-11 | 5 | 15 |
| 1985-01 | PHP - Frameworks | P016 | Rosas Ayala | Dario | NULL | NULL | NULL | Ju 19-22 y Sa 08-11 | 12 | 8 |

- 2) Implemente y explique un Script para crear un **procedimiento almacenado** para insertar datos a su base de datos.

```

EXEC sp_InsertarAlumno
'A003',
'Lopez',
'Carlos',
'Calle Ejemplo 123',
'999888777',
'carlos.lopez@example.com',
'miClaveSegura';

EXEC sp_InsertarAlumno
'A003',
'Rojas',
'Carlos',
'Calle Ejemplo 123',
'999888777',
'carlos.lopez@example.com',
'miClaveSegura';

select * from Alumno;

```

100 %

Results Messages

| | IdAlumno | ApeAlumno | NomAlumno | DirAlumno | TelAlumno | EmailAlumno | Clave |
|----|----------|---------------------|-----------------|-------------|------------|------------------------|----------|
| 24 | A0024 | Valverde Jaramillo | Saul | NULL | NULL | svalverde@hotmail.c... | 123456 |
| 25 | A0025 | Cipriano Avila | Roxana | NULL | NULL | rcipriano@hotmail.com | 123456 |
| 26 | A0026 | Rodriguez Quispe | Luis Alberto | NULL | NULL | lrodriguez@peru.com | 123456 |
| 27 | A0027 | Huerta Leon | Marco Antonio | NULL | NULL | mhuerta@gmail.com | 123456 |
| 28 | A0028 | Ortiz Fuentes | Ana Maria | NULL | NULL | aortiz@yahoo.es | 123456 |
| 29 | A0029 | Rivera Jaramillo | Martha | NULL | NULL | mrivera@gmail.com | 123456 |
| 30 | A003 | Lopez | Carlos | Calle Ej... | 9998887... | carlos.lopez@exampl... | miCla... |
| 31 | A0030 | Bustamante Camp... | Guino | NULL | NULL | gbustamante@hotmail... | 123456 |
| 32 | A0031 | Barreto Rodriguez | Carlos Miguel | NULL | NULL | cbarreto@gmail.com | 123456 |
| 33 | A0032 | Borja Gutti | José Miguel | NULL | NULL | jborja@uni.edu.pe | 123456 |
| 34 | A0033 | Espinoza De La C... | Ronny Cesar | NULL | NULL | respinoza@hotmail.com | 123456 |
| 35 | A0034 | Gómez Cáceres | Pedro Pablo | NULL | NULL | pgomez@peru.com | 123456 |
| 36 | A0035 | Gómez Manzano | José Alfonso | NULL | NULL | jgomez@elcomercio.pe | 123456 |
| 37 | A0036 | Herrera Farfán | Carlos Alberto | NULL | NULL | cherrera@peru.com | 123456 |
| 38 | A0037 | Huamanyauri Cuel... | David Roger | NULL | NULL | dhuamanyauri@ipae.pe | 123456 |
| 39 | A0038 | La Pierre Gutiérrez | Nathalie Aurora | NULL | NULL | nlapiere@telefonica.pe | 123456 |
| 40 | A0039 | Lema Gutiérrez | Katia | NULL | NULL | klema@ibm.com.pe | 123456 |
| 41 | A0040 | Leon Castro | Cesar Rómulo | NULL | NULL | cleon@peru.com | 123456 |
| 42 | A0041 | Manzanal Castro | Antonio Javier | NULL | NULL | amanzanal@gmail.com | 123456 |

Query executed successfully

- 3) Implemente y explique un Script para crear un **procedimiento almacenado** para eliminar datos a su base de datos

Mg. Ing. Raúl Fernández Bejarano

```

-- Crear el procedimiento almacenado
CREATE PROCEDURE sp_EliminarAlumno (
    @IdAlumno CHAR(5) -- Parámetro para identificar al alumno a eliminar
)
AS
BEGIN
    -- Verificar si el alumno existe antes de eliminar
    IF EXISTS (SELECT 1 FROM Alumno WHERE IdAlumno = @IdAlumno)
    BEGIN
        DELETE FROM Alumno WHERE IdAlumno = @IdAlumno;
        PRINT 'Alumno eliminado correctamente.';
    END
    ELSE
    BEGIN
        PRINT 'El alumno con el Id proporcionado no existe.';
    END
END;
GO
EXEC sp_EliminarAlumno 'A003';

```

IO %

Messages

(1 row affected)
 Alumno eliminado correctamente.

Completion time: 2024-11-21T16:59:37.2192831-05:00

- 4) Implemente y explique un Script para crear un **procedimiento almacenado** para actualizar datos a su base de datos

```

UPDATE Alumno
SET
    ApeAlumno = COALESCE(@ApeAlumno, ApeAlumno), -- Solo actualiza si se pasa un valor
    NomAlumno = COALESCE(@NomAlumno, NomAlumno),
    DirAlumno = COALESCE(@DirAlumno, DirAlumno),
    TelAlumno = COALESCE(@TelAlumno, TelAlumno),
    EmailAlumno = COALESCE(@EmailAlumno, EmailAlumno)
WHERE IdAlumno = @IdAlumno;

PRINT 'Alumno actualizado correctamente.';
END
ELSE
BEGIN
    PRINT 'El alumno con el Id proporcionado no existe.';
END
END;
GO
EXEC sp_ActualizarAlumno
    @IdAlumno = 'A0001',
    @ApeAlumno = 'Donayre Pérez',
    @EmailAlumno = 'cdonayre_updated@peru.com';

```

IO %

Messages

(1 row affected)
 Alumno actualizado correctamente.

Completion time: 2024-11-21T17:05:33.5229771-05:00

EXEC sp_ActualizarAlumno
 @IdAlumno = 'A0001',
 @ApeAlumno = 'Donayre Pérez',
 @DirAlumno = 'Av Solcito, 123',
 @EmailAlumno = 'cdonayre_updated@peru.com';

SELECT* FROM Alumno;

100 %

Results Messages

| | IdAlumno | ApeAlumno | NomAlumno | DirAlumno | TelAlumno | EmailAlumno | Clave |
|---|----------|------------------|------------|-----------------|-----------|---------------------------|--------|
| 1 | A0001 | Donayre Pérez | Christian | Av Solcito, 123 | NULL | cdonayre_updated@peru.com | 123456 |
| 2 | A0002 | Ortiz Rodriguez | Freddy | NULL | NULL | fortiz@hotmail.com | 123456 |
| 3 | A0003 | Silva Mejia | Ruth Ketty | NULL | NULL | ksilva@gmail.com | 123456 |
| 4 | A0004 | Melendez Noriega | Liliana | NULL | NULL | lmelendez@uni.edu.pe | 123456 |
| 5 | A0005 | Huerta Leon | Silvia | NULL | NULL | shuerta@peru.com | 123456 |

- 5) Implemente y explique un Script para crear un **procedimiento almacenado** para realizar cálculos matemáticos de una columna de su base de datos.

SQLQuery1.sql - LAB.../ (88) Executing... EduiecCreaBD.SQL...PC30(USER 17 (66))

```

OPEN cur;
FETCH NEXT FROM cur INTO @ExaParcial, @ExaFinal, @IdCursoProg, @IdAlumno;

-- Calcular el promedio para cada fila y actualizar la columna Promedio
WHILE @@FETCH_STATUS = 0
BEGIN
    -- Calcular el promedio de las calificaciones
    SET @Promedio = (@ExaParcial + @ExaFinal) / 2;

    -- Actualizar el valor del promedio en la tabla Matricula
    UPDATE dbo.Matricula
    SET Promedio = @Promedio
    WHERE IdCursoProg = @IdCursoProg AND IdAlumno = @IdAlumno;

    -- Obtener el siguiente registro
    FETCH NEXT FROM cur INTO @ExaParcial, @ExaFinal, @IdCursoProg, @IdAlumno;
END;

CLOSE cur;
DEALLOCATE cur;

PRINT 'Cálculos de promedio completados con éxito.';
END;
GO

EXEC sp_CalcularPromedio;

```

100 %

Results Messages

- 6) Implemente y explique un Script para crear un **disparador** para ingresar un registro automáticamente en una tabla de su base de datos.

Mg. Ing. Raúl Fernández Bejarano

```

SELECT
    IdCursoProg, -- ID del curso programado insertado
    CONCAT('Nuevo curso programado: ', IdCurso, ' para el ciclo ', IdCiclo) -- Descripción
FROM INSERTED; -- INSERTED contiene las filas insertadas
END;
GO

SELECT * FROM dbo.AuditoriaCursoProgramado;

```

| IdAuditoria | IdCursoProg | FechaRegistro | Descripcion |
|-------------|-------------|---------------|-------------|
|-------------|-------------|---------------|-------------|

- 7) Implemente y explique un Script para crear un **disparador** para elimine un registro automáticamente en una tabla de su base de datos.
- 8) Implemente y explique un Script para crear un **disparador** para actualice un registro automáticamente en una tabla de su base de datos.
- 9) Implemente y explique un Script para crear un **disparador** para verificar el control de datos (Ejemplo: que la nota ingresada este entre 0 y 20)
- 10) Utilizando Script Crear 03 usuarios con nombres de sus compañeros y uno suyo
- 11) Utilizando un script, copiar la base de datos (creada anteriormente) y compartir en cada uno de los usuarios
- 12) Utilizando un script, generar una copia de seguridad de la base de datos y compartir a cada uno de los usuarios
- 13) Utilizando un script, encriptar una de las tablas para que no se puedan ver los datos
- 14) Utilizando un script, aplique la seguridad a nivel de columna, restringiendo el acceso a la columna que contiene la clave primaria de una de las tablas de su base de datos
- 15) Utilizando un script, implementé seguridad a nivel de columna restringiendo el acceso a una de las columnas de una tabla.
- 16) Utilizando un script, realice el cifrado transparente de datos (TDE) para una las tablas.
- 17) 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.
- 18) Utilizando un Scripts realice la validación y filtración de entradas del usuario para evitar caracteres maliciosos (Ejemplo: ', --, ;)
- 19) Realice un script que verifiquen que los datos ingresados cumplan con formatos esperados (ej.: números en lugar de texto, longitud máxima).
- 20) Utilizando un script, configure la auditoría para el seguimiento y registro de acciones en la base de datos
- 21) Utilizando un script, configure de la memoria y el disco duro
- 22) Utilizando un script, genere una copia de seguridad de la base de datos
- 23) Realice un script para programar backups automatizados de su base de datos

```
EXEC msdb.dbo.sp_add_jobstep
    @job_name = 'Backup_NombreBaseDeDatos',
    @step_name = 'BackupPaso',
    @subsystem = 'TSQL',
    @command = 'BACKUP DATABASE NombreBaseDeDatos TO DISK = 'C:\Backups\NombreBaseDeDatos.bak';'

-- Agregar un horario para el job (diario a las 2 AM)
EXEC msdb.dbo.sp_add_schedule
    @schedule_name = 'BackupHorario',
    @enabled = 1,
    @freq_type = 4, -- Diario
    @freq_interval = 1, -- Todos los días
    @active_start_time = 20000; -- 2:00 AM

-- Asociar el job con el horario
EXEC msdb.dbo.sp_attach_schedule
    @job_name = 'Backup_NombreBaseDeDatos',
    @schedule_name = 'BackupHorario';

-- Habilitar el job
EXEC msdb.dbo.sp_add_jobserver
    @job_name = 'Backup_NombreBaseDeDatos';
```

100 %

Results Messages

- 24) Utilizando un script, genere la restauración de la base de datos
- 25) Utilizando un script, cree un espejo de la base de datos
- 26) Utilizando un script, para enviar datos a la base de datos espejo creada
- 27) Utilizando un script, de permiso a un usuario por un determinado tiempo
- 28) Utilizando un script, realice la replicación de bases de datos
- 29) Explique que es Always On Availability Groups
- 30) Explique que es Log Shipping