

UNIVERSIDAD ARGENTINA DE LA EMPRESA
Facultad de Ingeniería y Ciencias Exactas
INGENIERIA DE DATOS I

SIMULA FINAL

Docentes:

Ing. Juan C. Montero
Ing. Julieta Viarengo

Estudiante:

Legajo:

Fecha:

PC:

NOTA:

CASO PRÁCTICO

Parte 1 – Contexto del caso

La empresa **VAR Analytics S.A.** desarrolla un sistema para analizar estadísticas de la Liga Profesional de Fútbol Argentina (LPF).

El sistema debe permitir:

- Registrar los partidos oficiales.
- Registrar los jugadores de cada club.
- Registrar los eventos ocurridos en los partidos (goles, tarjetas, cambios).
- Validar reglas de negocio y registrar errores mediante triggers.
- Calcular resúmenes de estadísticas usando cursores y funciones.
- Exponer consultas de análisis mediante vistas.

A partir de este contexto, se propone el siguiente examen final.

INDICACIONES GENERALES

1. Trabaje en **SQL Server**.
2. Ejecute el script inicial provisto por la cátedra (archivo .txt o .sql) que:
 - Crea la base de datos: **VAR_Analytics_LPF**.
 - Crea las tablas: *Partido*, *Jugador*, *EventoPartido*.
 - Inserta datos de ejemplo (DML inicial).
3. No modifique los nombres de la base ni de las tablas ya creadas.
4. Todos los objetos que cree (tablas, funciones, procedimientos, triggers, vistas) deben hacerse dentro de la base **VAR_Analytics_LPF**.
5. Guarde su solución en un archivo **.sql** con su **apellido_nombre_legajo.sql**.

PARTE I – PRÁCTICA EN NOTEBOOK (T-SQL)

Ejercicio 1 – Tabla de alertas

Consigna:

Crear una tabla llamada **AlertaEventoPartido** que permita registrar intentos inválidos de inserción en la tabla *EventoPartido*. La tabla debe contener como mínimo los siguientes campos:

UNIVERSIDAD ARGENTINA DE LA EMPRESA
Facultad de Ingeniería y Ciencias Exactas
INGENIERIA DE DATOS I

- idAlerta → entero, clave primaria, autoincremental (IDENTITY).
- fechaHora → fecha y hora en que se produjo el error.
- idPartido → identificador del partido involucrado (puede ser NULL).
- idJugador → identificador del jugador involucrado (puede ser NULL).
- detalleError → texto descriptivo de la causa del error.

Ejercicio 2 – Procedimiento almacenado `spRegistrarEvento`

Consigna:

Crear un **procedimiento almacenado** llamado `spRegistrarEvento` que:

1. Reciba como parámetros:
 - @idPartido INT
 - @idJugador INT
 - @minuto TINYINT
 - @tipoEvento VARCHAR(10)
 - @descripcion VARCHAR(100) = NULL (parámetro opcional)
2. Valide dentro del procedimiento:
 - Que el partido (@idPartido) exista en la tabla Partido.
 - Que el jugador (@idJugador) exista en la tabla Jugador.

Si alguno no existe, debe lanzar un error utilizando THROW.

3. Genere un nuevo idEvento de manera **manual**, calculando $\text{MAX}(\text{idEvento}) + 1$ sobre la tabla EventoPartido.
4. Realice el INSERT en la tabla EventoPartido.
5. Use un bloque **TRY-CATCH** para capturar errores, y dentro del CATCH:
 - Obtener información del error (mensaje, número, estado) con las funciones de error (ERROR_MESSAGE(), ERROR_NUMBER(), ERROR_STATE()).
 - Re-lanzar el error hacia el cliente usando THROW.

Nota: El procedimiento **no** debe insertar registros en **AlertaEventoPartido**. La responsabilidad de registrar alertas será del trigger del **Ejercicio 3**.

Ejercicio 3 – Trigger `trg_ValidarEventoPartido`

Consigna:

Crear un **trigger** llamado **`trg_ValidarEventoPartido`** sobre la tabla **EventoPartido**, que se dispare **después de un INSERT** y que:

1. Valide que:
 - El campo minuto esté entre 1 y 90.
 - El campo tipoEvento sea uno de: 'GOL', 'AMARILLA', 'ROJA', 'CAMBIO'.
 - El jugador (idJugador) pertenezca al equipo local o visitante del partido correspondiente (equipoLocal o equipoVisitante de la tabla Partido).

UNIVERSIDAD ARGENTINA DE LA EMPRESA
Facultad de Ingeniería y Ciencias Exactas
INGENIERIA DE DATOS I

2. Si alguna de estas reglas se viola, el trigger debe:
- Insertar un registro en la tabla **AlertaEventoPartido** indicando *fechaHora*, *idPartido*, *idJugador* y un *detalleError* descriptivo.
 - Cancelar la operación de inserción en **EventoPartido** mediante ROLLBACK TRANSACTION.
 - Lanzar un error con RAISERROR apropiado para informar al cliente que el evento no pudo registrarse.

Ejercicio 4 – Función fn_TotalGolesJugador

Consigna:

Crear una función llamada **fn_TotalGolesJugador** que:

- Reciba como parámetro @idJugador INT.
- Calcule cuántos eventos con tipoEvento = 'GOL' tiene ese jugador en la tabla **EventoPartido**.
- Devuelva ese valor como un entero (INT).
- Si el jugador no tiene ningún gol registrado, la función debe devolver **0**.

Pregunta 1 – SP y Trigger

El SP spRegistrarEvento realiza un INSERT sobre EventoPartido, y existe un trigger AFTER INSERT sobre esa tabla.

Afirmaciones:

- A. El INSERT dentro del SP dispara el trigger AFTER INSERT.
- B. Si el trigger ejecuta un ROLLBACK, el evento no se inserta.
- C. El SP puede capturar el error del trigger mediante TRY-CATCH.
- D. El SP siempre registra los errores en AlertaEventoPartido.

Opciones:

- 1. Solo A y B
- 2. Solo A, B y C
- 3. Solo C y D
- 4. Todas son correctas

Pregunta 2 – Trigger e inserted

Afirmaciones:

- A. La tabla lógica inserted puede contener más de una fila.
- B. El trigger se ejecuta una vez por sentencia INSERT, no una vez por fila.
- C. Si se ejecuta ROLLBACK TRANSACTION dentro del trigger, se revierten todas las filas insertadas por esa sentencia.
- D. El trigger puede usar JOIN entre inserted, Partido y Jugador.

UNIVERSIDAD ARGENTINA DE LA EMPRESA
Facultad de Ingeniería y Ciencias Exactas
INGENIERIA DE DATOS I

Opciones:

1. Solo A y B
2. Solo B y C
3. Solo A, B y C
4. Todas son correctas

Pregunta 5 – Cardinalidad Partido–Evento–Jugador

Afirmaciones:

- A. La relación entre Partido y EventoPartido es de 1 a N.
- B. La relación entre Jugador y EventoPartido es de 1 a N.
- C. Conceptualmente hay una relación muchos a muchos entre Partido y Jugador a través de EventoPartido.
- D. Con un JOIN entre las tres tablas se puede obtener la lista de jugadores que participaron en un partido.

Opciones:

1. Solo A y B
2. Solo A, B y D
3. Solo C y D
4. Todas son correctas

Pregunta 9 – 1FN (PartidoPrensa)

Relación:

```
PartidoPrensa(  
    idPartido,  
    equiposInvolucrados,  
    fotografosAsignados  
)
```

idPartido	equiposInvolucrados	fotografosAsignados
1	'Boca Juniors / River Plate'	'Pérez; Gómez; Suárez'

Afirmaciones:

- A. equiposInvolucrados viola 1FN por contener valores multivaluados.
- B. fotografosAsignados viola 1FN por contener una lista de valores.
- C. Para cumplir 1FN se deberían crear tablas separadas para los fotógrafos y para los equipos.
- D. El uso de tipo VARCHAR garantiza automáticamente el cumplimiento de 1FN.

Opciones:

1. Solo A y B
2. Solo A, B y C
3. Solo C y D
4. Todas son correctas

UNIVERSIDAD ARGENTINA DE LA EMPRESA
Facultad de Ingeniería y Ciencias Exactas
INGENIERIA DE DATOS I

Pregunta 10 – 2FN (JugadorPartido)

Relación:

```
JugadorPartido(  
    idJugador,  
    idPartido,  
    nombreJugador,  
    equipoJugador,  
    minutosJugados  
)  
Clave primaria: (idJugador, idPartido)
```

Afirmaciones:

- A. nombreJugador depende solo de idJugador.
- B. equipoJugador depende solo de idJugador.
- C. minutosJugados depende de la combinación (idJugador, idPartido).
- D. Existe una violación de 2FN por dependencias parciales.

Opciones:

- 1. Solo A y B
- 2. Solo A, B y C
- 3. A, B, C y D
- 4. Solo C y D

Pregunta 11 – 3FN (PartidoExtendido)

Relación:

```
PartidoExtendido(  
    idPartido,  
    fecha,  
    idEstadio,  
    nombreEstadio  
)  
Clave: idPartido
```

Afirmaciones:

- A. nombreEstadio depende de idEstadio.
- B. Existe una dependencia transitiva $\text{idPartido} \rightarrow \text{idEstadio} \rightarrow \text{nombreEstadio}$.
- C. Para 3FN conviene separar una tabla Estadio(idEstadio, nombreEstadio).
- D. En 3FN todo atributo no clave debe depender directamente de la clave primaria sin dependencias transitivas.

Opciones:

- 1. Solo A y B
- 2. Solo C y D
- 3. Solo A, B y C
- 4. Todas son correctas

UNIVERSIDAD ARGENTINA DE LA EMPRESA
Facultad de Ingeniería y Ciencias Exactas
INGENIERIA DE DATOS I

Pregunta 12 – 4FN y 5FN (Jugador–Club–Sponsor)

Relación:

```
JugadorClubSponsor(  
    idJugador,  
    idClub,  
    idSponsor  
)
```

Hechos:

- Un jugador puede jugar en varios clubes a lo largo de su carrera.
- Un jugador puede tener varios sponsors personales.
- Los sponsors están vinculados al jugador, no directamente al club.

Afirmaciones:

- A. Existen dos dependencias multivaluadas independientes: Jugador–Club y Jugador–Sponsor.
- B. Esta situación puede violar la 4ª Forma Normal (4FN).
- C. Una solución es descomponer en JugadorClub(idJugador, idClub) y JugadorSponsor(idJugador, idSponsor).
- D. Para 5ª Forma Normal (5FN) se debe analizar si la descomposición permite reconstruir la relación original sin pérdida mediante joins necesarios.

Opciones:

1. Solo A y B
2. Solo A, B y C
3. Solo C y D
4. Todas son correctas

Pregunta 13 – Razonamiento: evento inválido

Se ejecuta el siguiente comando:

```
EXEC spRegistrarEvento  
    @idPartido = 1,      -- Partido: Boca Juniors vs River Plate  
    @idJugador = 40,    -- Jugador: Adam Bareiro (San Lorenzo)  
    @minuto = 25,  
    @tipoEvento = 'GOL',  
    @descripcion = 'Gol de Bareiro';
```

El sistema muestra un mensaje de error y, al consultar la tabla **AlertaEventoPartido**, aparece un registro que indica que el equipo del jugador está fuera del partido.

Consigna:

Explique, en forma sintética y ordenada, qué ocurre internamente en:

- el procedimiento almacenado spRegistrarEvento,
- el trigger trg_ValidarEventoPartido,
- y las tablas EventoPartido y AlertaEventoPartido,

para que el evento **no quede registrado** en **EventoPartido** pero sí aparezca una alerta en **AlertaEventoPartido**.

PLANTEAMIENTO MODELO CON PREGUNTAS DEL CÓDIGO

Tenemos el siguiente código:

```
CREATE OR ALTER FUNCTION fn_TotalGolesJugador
(
    @idJugador INT
)
RETURNS INT
AS
BEGIN
    DECLARE @total INT;

    SELECT @total = COUNT(*)
    FROM EventoPartido
    WHERE idJugador = @idJugador
    AND tipoEvento = 'GOL';

    RETURN ISNULL(@total, 0);
END;
GO
```

1. ¿Cuál de las siguientes afirmaciones es correcta respecto del uso de la función **fn_TotalGolesJugador**?

- A. La función puede usarse dentro de un SELECT para mostrar la cantidad total de goles de un jugador.
- B. La función puede modificar datos en la tabla EventoPartido.
- C. La función debe devolver NULL cuando el jugador no tiene goles.
- D. La función requiere obligatoriamente un TRY-CATCH interno.

2. En la función **fn_TotalGolesJugador**, el uso de **ISNULL(@total, 0)** implica que:

- A. La función devuelve 0 si no existen registros para ese jugador con tipoEvento = 'GOL'.
- B. La función lanza un error si no encuentra goles.
- C. La función asume que @total siempre tendrá un valor inicial de 0.
- D. La función devuelve NULL si @total no fue asignado.

Dado el modelo:

- EventoPartido(idEvento, idPartido, idJugador, minuto, tipoEvento, descripcion)
- Jugador(idJugador, nombre, apellido, equipo)

UNIVERSIDAD ARGENTINA DE LA EMPRESA
Facultad de Ingeniería y Ciencias Exactas
INGENIERIA DE DATOS I

3. ¿Qué garantiza que la función `fn_TotalGolesJugador` pueda obtener correctamente la cantidad de goles de un jugador?

- A. Que exista una clave foránea desde `EventoPartido` hacia `Jugador`.
- B. Que `tipoEvento` esté validado por un `CHECK` para permitir solo valores correctos.
- C. Que la función realice `JOIN` con la tabla `Partido`.
- D. Que la función utilice `COUNT(*)` en lugar de `SUM`.

4. La empresa VAR Analytics te solicita modificar el sistema para que, además del total de goles, la función devuelva también el nombre completo del jugador y su club actual. ¿Es posible utilizar una función `ESCALAR` para cumplir este nuevo requerimiento? Justifique técnica y conceptualmente su respuesta.

En tu justificación, ten en cuenta:

- qué tipo de dato devuelve una función `escalar`,
- si puede o no devolver múltiples valores,
- qué alternativas existen en T-SQL para devolver más de un atributo,
- y cómo impactaría cambiar la función en el diseño del modelo.

5. Uso y comportamiento de la función

Afirmaciones:

- A. La función puede usarse en una cláusula `SELECT` para devolver el total de goles de un jugador.
- B. La función puede devolver `NULL` si el jugador no tiene goles registrados.
- C. La función puede ser llamada dentro de un `WHERE` de una consulta.
- D. La función puede modificar datos en la tabla `EventoPartido`.

Opciones:

- a. Solo A y C
- b. Solo A, B y C
- c. Solo B y D
- d. Todas son correctas

6. Comprensión interna de la función

Afirmaciones:

- A. La línea `@total = COUNT(*)` asigna el número total de filas que cumplen la condición.
- B. La función utiliza `ISNULL(@total, 0)` para evitar devolver un valor `NULL`.
- C. Si no hay goles registrados, `COUNT(*)` devuelve `NULL`.
- D. La función podría escribirse sin la variable `@total` usando directamente `RETURN (...)`.

Opciones:

- a. Solo A y B
- b. Solo A, B y D
- c. Solo C y D

- d. Todas son correctas

7. Relación entre modelo relacional y la función

Afirmaciones:

- A. Para que la función funcione correctamente, debe existir una clave foránea desde EventoPartido hacia Jugador.
- B. La función solo contabiliza goles si el tipoEvento está correctamente validado mediante el CHECK de la tabla.
- C. La función podría devolver resultados incorrectos si se registran eventos con valores de tipoEvento fuera del dominio permitido.
- D. La función requiere hacer JOIN con la tabla Partido para conocer los goles del jugador.

Opciones:

- a. Solo A y B
- b. Solo A, B y C
- c. Solo C y D
- d. Todas son correctas

NOTA: *para el final se debe justificar las respuestas.*

¡Éxitos!