

## Ejercicios de SQL: Triggers, Funciones y Procedimientos

### Dificultad: Básico

#### Ejercicio 1

Crea un procedimiento almacenado `spInsertarUsuario` que reciba el nombre, email, contraseña y edad de un usuario, e inserte un nuevo registro en la tabla `Users`.

#### Ejercicio 2

Crea un procedimiento almacenado `spInsertarPlataforma` que reciba el nombre de la plataforma y el valor para `isHandheld`. Antes de insertar, comprueba si ya existe otra plataforma con ese mismo nombre. Si existe, no insertes y devuelve un mensaje de error.

#### Ejercicio 3

Crea un trigger que, antes de insertar un nuevo videojuego (`Videogames`), compruebe que el precio sea siempre mayor o igual a 0. En caso contrario, impide la inserción.

#### Ejercicio 4

Crea una tabla `VideogamesAudit` y un trigger `tAuditVideogames` que, cada vez que se inserte o elimine un videojuego, registre en `VideogamesAudit` la operación realizada.

## Ejercicios de SQL: Triggers, Funciones y Procedimientos

### Dificultad: Intermedio

#### Ejercicio 1

Crea una función llamada `fnEdadPromedioUsuarios()` que devuelva la edad media de todos los usuarios. Ejecuta esta función en una sentencia `SELECT` para mostrar el resultado.

#### Ejercicio 2

Crea una función `fnEsPuntuacionValida(score DECIMAL(4,2))` que devuelva 1 si la puntuación es entre 0 y 10, y 0 en caso contrario. Úsala en un trigger para validar la inserción de nuevas reviews.

#### Ejercicio 3

Crea un procedimiento `spBorrarUsuarioYJuegos(userID INT)` que elimine un usuario y sus videojuegos en una transacción segura.

#### Ejercicio 4

Crea un procedimiento `spAsignarLogroATodos(idAchievement INT)` que asigne un logro específico a todos los usuarios sin duplicados.

## Ejercicios de SQL: Triggers, Funciones y Procedimientos

### Dificultad: Avanzado

#### Ejercicio 1

Crea un procedimiento `spRecalcularScores()` que recorra todos los videojuegos, calcule el promedio de las rating de Reviews y actualice el campo `score` en Videogames.

#### Ejercicio 2

Crea un trigger `tLogPrecioVideogame` para que, al actualizar el precio de un videojuego, registre la operación en una tabla `PriceChanges`.

#### Ejercicio 3

Crea una función `fnContarAmigos(idUser INT)` que devuelva la cantidad total de amigos de un usuario.

#### Ejercicio 4

Crea un procedimiento `spMostrarInformacionUsuario(idUser INT)` que muestre información del usuario, incluyendo el conteo de amigos y videojuegos.

## Ejercicios de SQL: Triggers, Funciones y Procedimientos

### Dificultad: Extra

#### Ejercicio 1

Crea un trigger tNotificarLogro que, al obtener un nuevo logro en UsersAchievements, inserte una notificación en una tabla Notifications.

#### Ejercicio 2

Crea un procedimiento spGenerarRankingMensual() que calcule la puntuación total de logros por usuario y registre el ranking mensual en una tabla.

#### Ejercicio 3

Crea una función fnPrecioTotalColeccion(idUser INT) que devuelva la suma de los precios de todos los videojuegos que posee un usuario.

#### Ejercicio 4

Crea un procedimiento spAsignarLogrosPorPuntuacion() que encuentre todos los videojuegos con puntuación perfecta y asigne un logro especial a los usuarios que los posean.