

TP: Repaso SP + Manejo de Errores y Transacciones

Recuerde Utilizar el script provisto...[LINK](#)

Ejercicio 1:

Crea un procedimiento almacenado (insertarEnTablaInexistente) que intente insertar un registro en una tabla inexistente y maneje el error con un handler SQLEXCEPTION.

Ejercicio 2:

Crea un procedimiento almacenado (crearTablaExistente) que intente crear una tabla que ya existe y maneje la advertencia con un handler SQLWARNING.

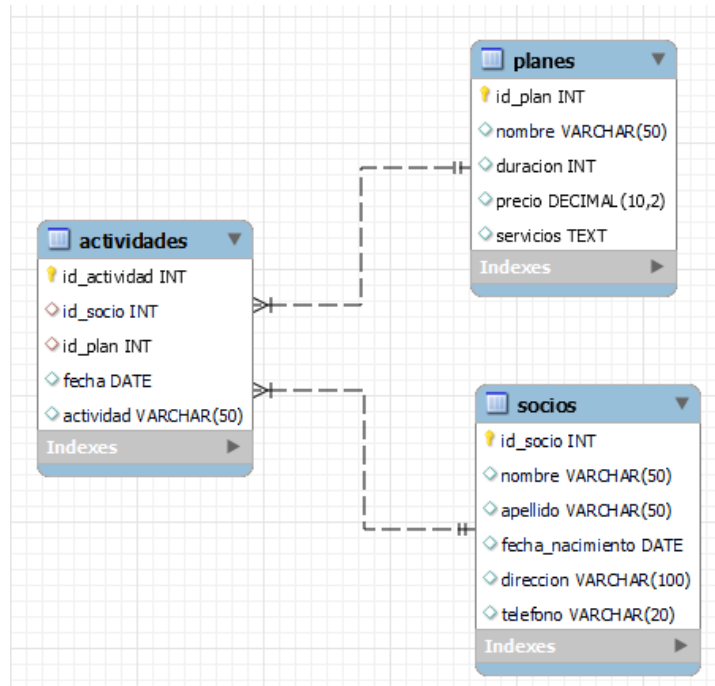
Ejercicio 3:

Crea un procedimiento almacenado (CREATE PROCEDURE seleccionarRegistroInexistente) que intente seleccionar un registro de una tabla basándose en un ID que no existe y maneje la condición NOT FOUND.

Ejercicio 4 (¿cómo funciona la selección de handlers?):

Crea un procedimiento almacenado (manejoCombinado) que intente insertar un registro en una tabla inexistente y luego intente crear una tabla que ya existe, manejando ambos casos con sus respectivos handlers.

Según el siguiente DER



Ejercicio 5:

Crea un procedimiento almacenado (insertarActividad) que intente insertar un registro en la tabla actividades y maneje el error si el id_socio o id_plan no existen en sus respectivas tablas.

Ejercicio 6:

Crea un procedimiento almacenado (seleccionarSocio) que intente seleccionar un registro de la tabla socios basándose en un ID que no existe y maneje la condición NOT FOUND.

Manejo de Transacciones

Ejercicio 7:

Desarrolla un procedimiento almacenado llamado registrarSocioConPlan. Este procedimiento debe recibir como

parámetros los datos de un nuevo socio (nombre, apellido, fecha_nacimiento, dirección, teléfono) y los IDs de un plan y una actividad. El procedimiento debe:

1. Insertar el nuevo socio en la tabla socios.
2. Insertar una nueva actividad para ese socio y el plan especificado en la tabla actividades.
3. Manejar los siguientes errores:
 - o Si ocurre un error durante la inserción del socio, realizar un ROLLBACK y devolver un mensaje de error.
 - o Si los IDs del plan o la actividad no existen, realizar un ROLLBACK y devolver un mensaje de error.
4. Si todas las operaciones se realizan con éxito, realizar un COMMIT y devolver un mensaje de éxito.

Ejercicio 8:

Desarrolla un procedimiento almacenado llamado actualizarPlanYRegistrarActividad. Este procedimiento debe recibir como parámetros el ID de un plan, su nuevo precio, el ID de un socio y el ID de una actividad. El procedimiento debe:

1. Actualizar el precio del plan especificado en la tabla planes.
2. Insertar una nueva actividad para el socio y el plan especificados en la tabla actividades.
3. Manejar los siguientes errores:
 - o Si ocurre un error durante la actualización del plan o la inserción de la actividad, realizar un ROLLBACK y devolver un mensaje de error.
 - o Si los IDs del plan, socio o actividad no existen, realizar un ROLLBACK y devolver un mensaje de error.
4. Si todas las operaciones se realizan con éxito, realizar un COMMIT y devolver un mensaje de éxito.

Ejercicio 9:

Desarrolla un procedimiento almacenado llamado `eliminarSocioYActividades`. Este procedimiento debe recibir como parámetro el ID de un socio. El procedimiento debe:

1. Eliminar todas las actividades asociadas al socio en la tabla `actividades`.
2. Eliminar el socio de la tabla `socios`.
3. Manejar los siguientes errores:
 - o Si ocurre un error durante la eliminación de las actividades o del socio, realizar un `ROLLBACK` y devolver un mensaje de error.
 - o Si el socio no existe, realizar un `ROLLBACK` y devolver un mensaje de error.
4. Si todas las operaciones se realizan con éxito, realizar un `COMMIT` y devolver un mensaje de éxito.