

ETAPA 2 - Informe Individual

Jóvenes a Programar (JaP)

Proyecto TESTING 2024 | 18 de Octubre 2024 | Grupo 298

Referente: Paula Rodríguez

Tutor: Carlos Da Rosa



Alumna:

Herrero Florencia

Introducción:	3
Alcance:	4
Desarrollo:	5
Etapa de pruebas:	10
CP48:	10
Resultado:	10
CP49:	11
Resultado:	12
CP50:	12
Resultado:	13
CP51:	14
Resultado:	15
CP52:	16
Resultado:	16
CP53:	17
Resultado:	18
CP54:	18
Resultado:	19
CP55:	19
Resultado:	20
CP56:	20
Resultado:	21
CP57:	21
Resultado:	22
CP58:	23
Resultado:	24
CP59:	24
Resultado:	25
CP61:	26
Resultado:	26
Conclusión:	28
Anexos:	29

Introducción:

Este informe detalla los casos de prueba diseñados para evaluar el software de la veterinaria *Guau Guau*. El objetivo principal de este sistema es permitir el registro y la gestión de la información de los clientes y sus respectivas mascotas, con el fin de ofrecer un servicio de mayor calidad. A través de estos casos de prueba, se busca identificar, analizar y procesar los datos obtenidos, asegurando que coincidan con la información proporcionada por el cliente y el documento ESRE.

Alcance:

El presente informe detalla el alcance de las pruebas ejecutadas en la base de datos para verificar la correcta implementación de las tablas y sus restricciones asociadas. Estas pruebas fueron diseñadas para garantizar que los datos almacenados en la base de datos cumplan con las reglas de integridad y consistencia definidas por el cliente, y que las restricciones SQL sean aplicadas correctamente en todos los escenarios previstos. Por otro lado se realiza una prueba de relación entre opciones de registros en Registro_cliente.html formato básico utilizado para estructurar páginas web. Este archivo contiene el código que define cómo se verá una página web cuando se abra en un navegador.

Tipos de pruebas realizadas:

Pruebas de validación de restricciones: Verificación de que todas las restricciones definidas en la base de datos (como FOREIGN KEY, UNIQUE, NOT NULL, CHECK) estén funcionando correctamente y generen los errores adecuados cuando se violen.

Pruebas de consistencia: Ejecución de transacciones para comprobar que las relaciones entre las tablas se mantienen consistentes después de las inserciones, actualizaciones o eliminaciones de datos.

Pruebas de validación de datos duplicados: Comprobación de que las restricciones de unicidad y las claves primarias previenen la inserción de registros duplicados.

Entorno de pruebas:

- **Dispositivo:** PC-Laptop (ceibalita) con procesador Intel(R) Celeron(R) CPU N3160 1.60GHz y 4.00GB de RAM
- **Sistema Operativo:** Windows 10 Pro Education 22H2
- **Base de datos:** Microsoft SQL Server Management Studio
- **Versión:** 20.2.30.0
- **Navegador:** Google Chrome, Versión 130.0.6723.58 (Build oficial) (64 bits)

Exclusiones:

- **Pruebas de seguridad:** No se realizaron pruebas de seguridad para verificar vulnerabilidades como inyecciones SQL o acceso no autorizado a la base de datos.
- **Pruebas de rendimiento a gran escala:** No se ejecutaron pruebas de carga o rendimiento en grandes volúmenes de datos o en condiciones de alto tráfico de usuarios.

Desarrollo:

En esta sección se detallan las pruebas realizadas sobre la base de datos del sistema, centradas en validar la correcta implementación de las tablas y sus restricciones asociadas. Se llevaron a cabo pruebas de validación de restricciones, consistencia de datos y prevención de duplicados, todas con el objetivo de garantizar la integridad de los datos y el cumplimiento de las reglas definidas por el cliente. A continuación, se describen los casos de prueba ejecutados, los resultados obtenidos y los problemas encontrados durante las pruebas.

Casos de prueba a trabajar:

Tabla Clientes:

ID	DESCRIPCIÓN	PASOS A SEGUIR	RESULTADOS ESPERADOS
CP48	Intentar insertar cliente sin el campo Dirección	INSERT INTO Clientes (Nombre, Apellido, Email) VALUES ('Juan', 'Pérez', 'juan.perez@example.com');	Error de restricción NOT NULL.
CP49	Intentar insertar al cliente con un email que ya existe.	INSERT INTO Clientes (Nombre, Apellido, Email, Direccion) VALUES ('María', 'Gómez', 'juan.perez@example.com', 'Calle 456');	Error de restricción UNIQUE.
CP50	Actualizar la dirección de un cliente existente en la BBDD.	UPDATE Clientes SET Direccion = 'Nueva Calle 123' WHERE ClienteID = 1;	Se actualiza correctamente la dirección del cliente.
CP51	Eliminar un cliente que no tiene mascotas asociadas en la base de datos	DELETE FROM Clientes WHERE ClienteID = 2;	El cliente se elimina correctamente.

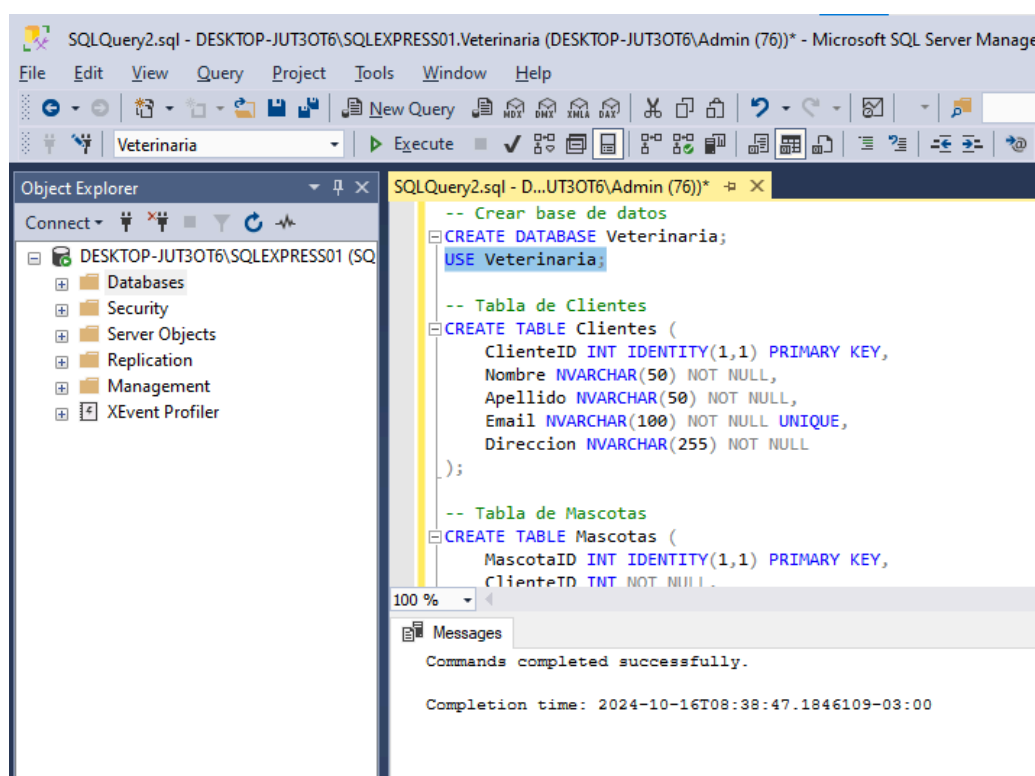
Tabla Usuarios:

ID	DESCRIPCIÓN	PASOS A SEGUIR	RESULTADOS ESPERADOS
CP52	Inserción de usuario con todos los datos válidos.	INSERT INTO Usuarios (NombreUsuario, Contraseña, Rol) VALUES ('admin', 'segura123', 'Administrador');	Inserción exitosa, se genera UsuarioID.
CP53	Intentar insertar usuario sin el campo NombreUsuario.	INSERT INTO Usuarios (Contraseña, Rol) VALUES ('segura123', 'Administrador');	Error de restricción NOT NULL.
CP54	Intentar insertar usuario sin el campo Contraseña	INSERT INTO Usuarios (NombreUsuario, Rol) VALUES ('admin', 'Administrador');	Error de restricción NOT NULL.
CP55	Intentar insertar un usuario con un nombre de usuario que ya existe en la BBDD	INSERT INTO Usuarios (NombreUsuario, Contraseña, Rol) VALUES ('admin', 'otra123', 'Empleado');	Error de restricción UNIQUE.
CP56	Intentar insertar usuario con un rol no permitido	INSERT INTO Usuarios (NombreUsuario, Contraseña, Rol) VALUES ('admin1', 'segura123', 'Gerente');	Error de restricción CHECK.
CP57	Eliminar un usuario existente de la BBDD	DELETE FROM Usuarios WHERE UsuarioID = 1;	El usuario se elimina correctamente de la tabla Usuarios.

Relaciones entre tablas:

ID	DESCRIPCIÓN	PASOS A SEGUIR	RESULTADOS ESPERADOS
CP58	Intentar eliminar un cliente que tiene mascotas registradas.	DELETE FROM Clientes WHERE ClienteID = 1;	Error de restricción FOREIGN KEY (no se permite eliminar).
CP59	Intentar eliminar una mascota que tiene vacunas registradas.	DELETE FROM Mascotas WHERE MascotaID = 1;	Error de restricción FOREIGN KEY (no se permite eliminar).

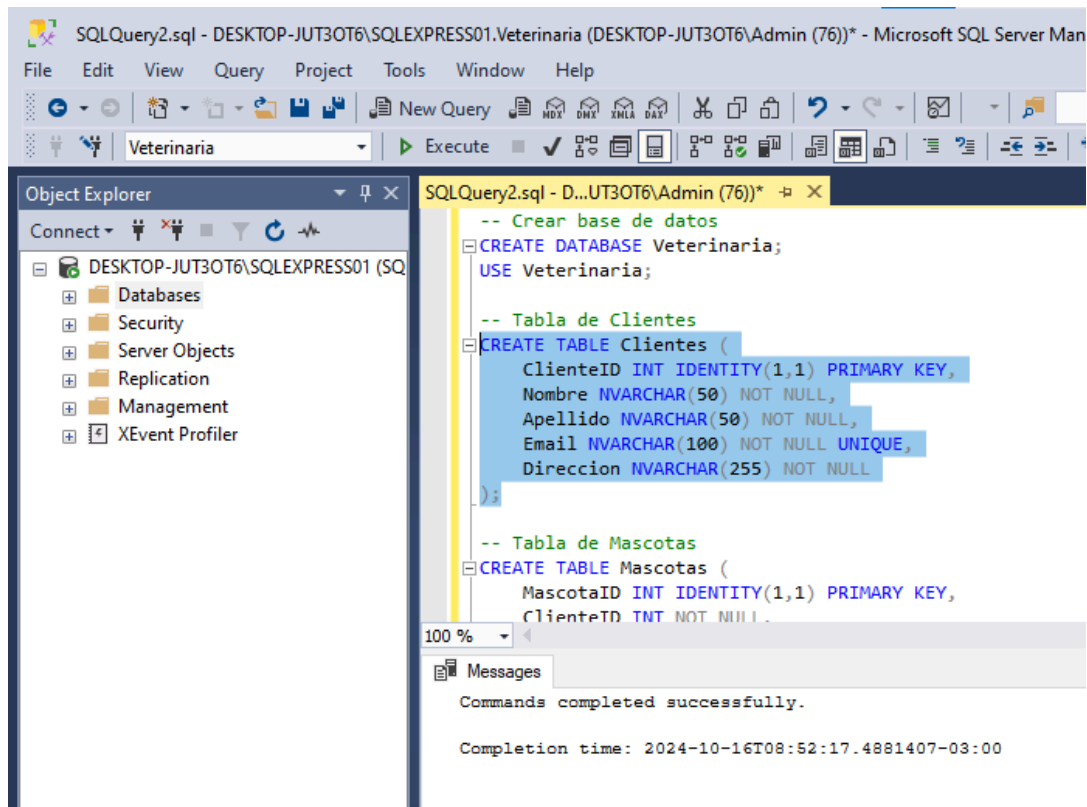
Para iniciar la etapa de pruebas, se accede al sistema Microsoft SQL Server, donde se ejecutarán las sentencias previamente definidas, las cuales se detallan a continuación. En primer lugar, se creará la base de datos, lo que permitirá la creación de las tablas y el desarrollo de los casos de prueba



(Figura 1 - Creando en SQL server la BBDD)

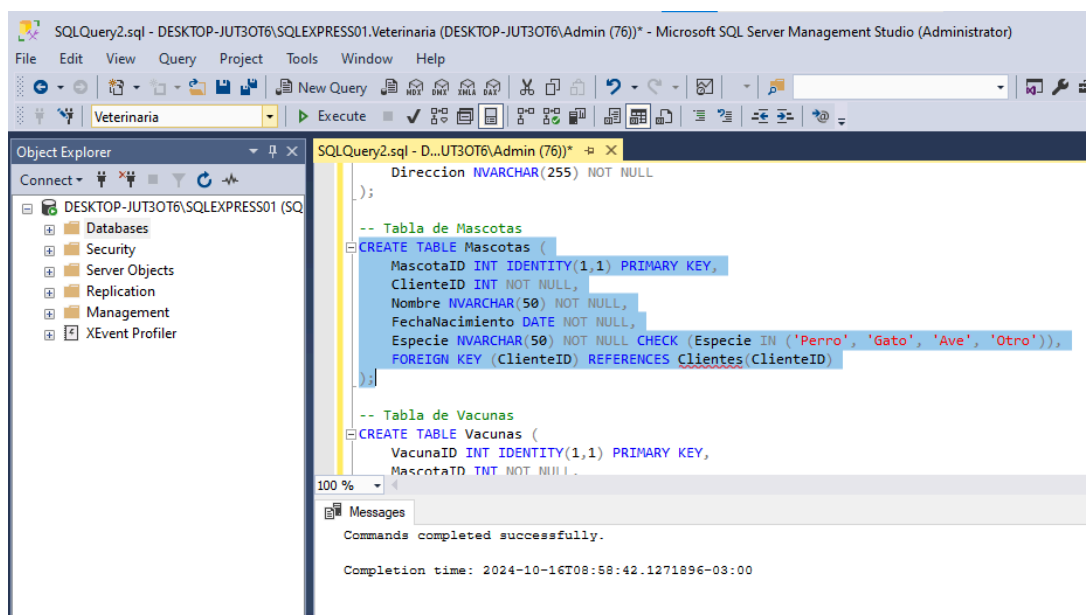
Una vez creada la base de datos, procederemos a la creación de las siguientes tablas, con el fin de cumplir con las precondiciones de los casos de prueba:

Tabla clientes:



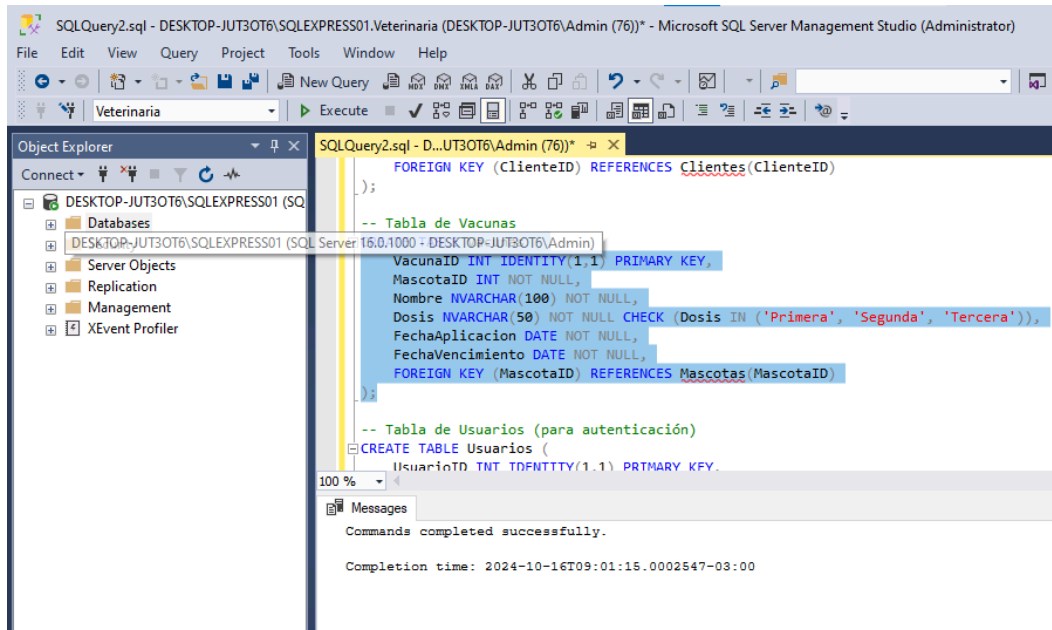
(Figura 2 - Tabla Clientes)

Tabla Mascota:



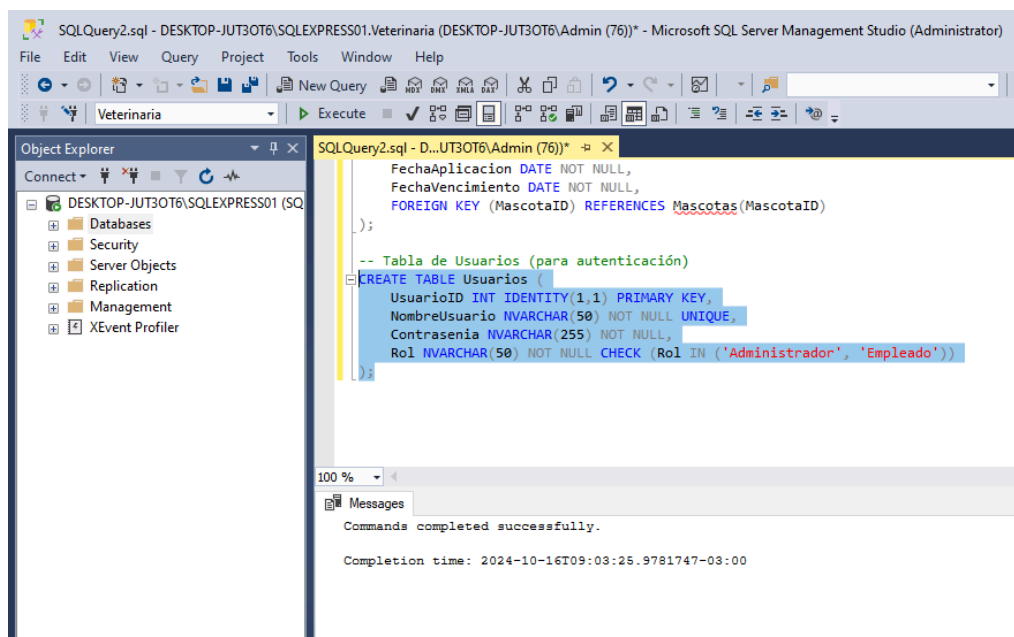
(Figura 3 - Tabla Mascota)

Tabla Vacunas:



(Figura 4 -Tabla Vacunas)

Tabla Usuarios:



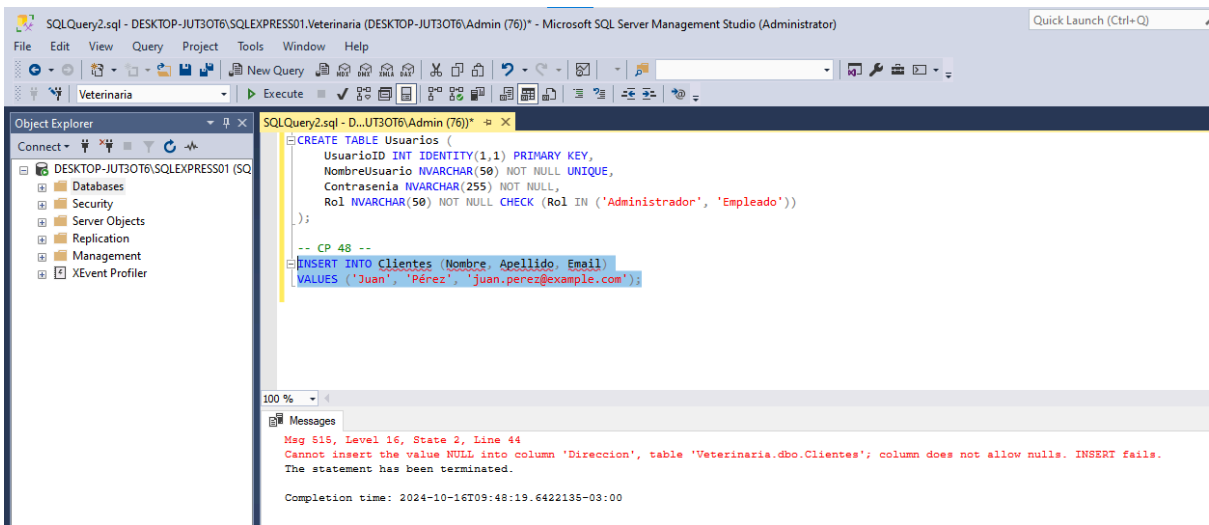
(Figura 5 - Tabla Usuarios)

Etapa de pruebas:

CP48:

ID	DESCRIPCIÓN	PASOS A SEGUIR	RESULTADOS ESPERADOS
CP48	Intentar insertar cliente sin el campo Dirección	INSERT INTO Clientes (Nombre, Apellido, Email) VALUES ('Juan', 'Pérez', 'juan.perez@example.com');	Error de restricción NOT NULL.

Para este caso de prueba las precondiciones son que la tabla Clientes ya esté creada en la BBDD (Figura 2). Se procede a insertar la sentencia para observar el resultado:



(Figura 6 - CP48)

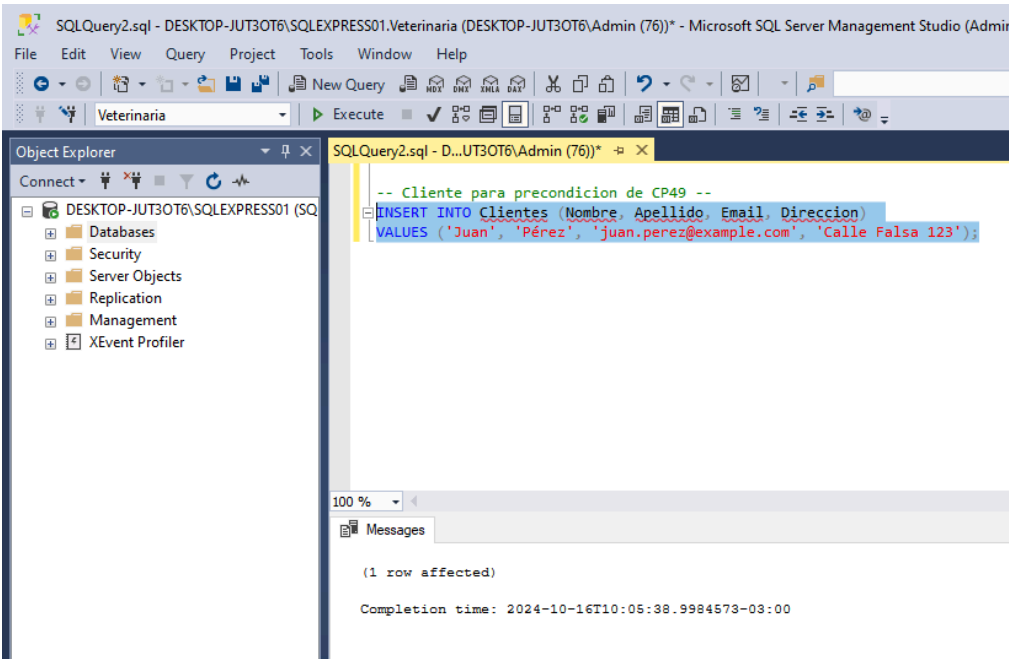
Resultado:

Como se puede observar en la Figura 6, al ejecutar la sentencia en SQL se obtiene la siguiente mensaje: "Msg 515, Nivel 16, Estado 2, Línea 44, No se puede insertar el valor NULL en la columna 'Direccion', de la tabla 'Veterinaria.dbo.Clientes'; la columna no permite valores nulos. La inserción falla coincidiendo con el resultado esperado, Se cambia estado a aprobado en tabla de control de casos de prueba (Ver anexo).

CP49:

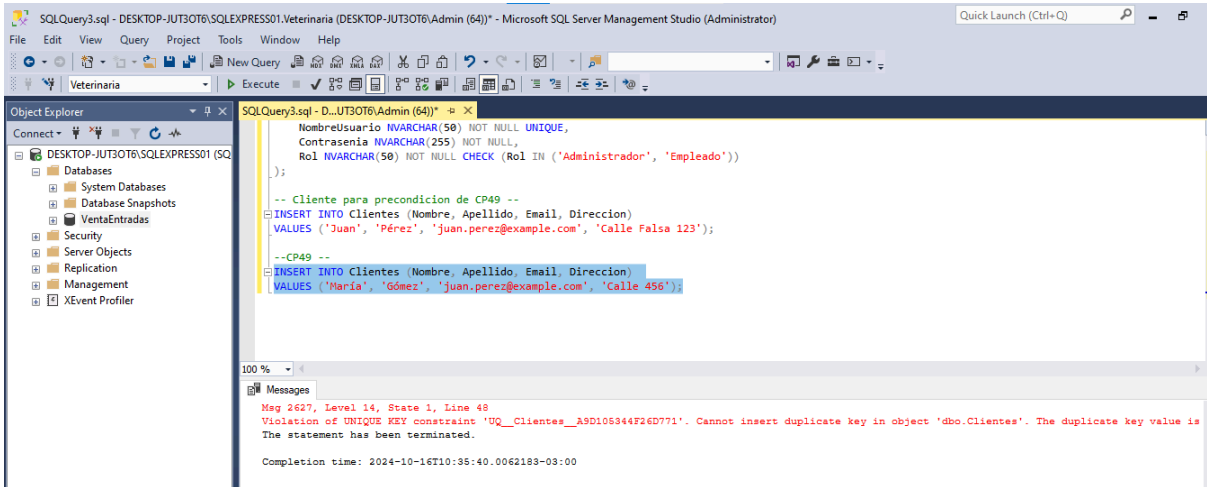
ID	DESCRIPCIÓN	PASOS A SEGUIR	RESULTADOS ESPERADOS
CP49	Intentar insertar al cliente con un email que ya existe.	INSERT INTO Clientes (Nombre, Apellido, Email, Direccion) VALUES ('María', 'Gómez', 'juan.perez@example.com', 'Calle 456');	Error de restricción UNIQUE.

Para este caso de prueba las precondiciones son que la tabla Clientes ya esté creada en la BBDD (Figura 2) y que exista un cliente registrado y validado con el email que intentamos validar, se procede a cargar en la BBDD dicho cliente para que se cumplan las precondiciones:



(Figura 7 - Carga de cliente para CP49)

Se procede a insertar la sentencia para verificar el resultado:



(Figura 8 - CP49)

Resultado:

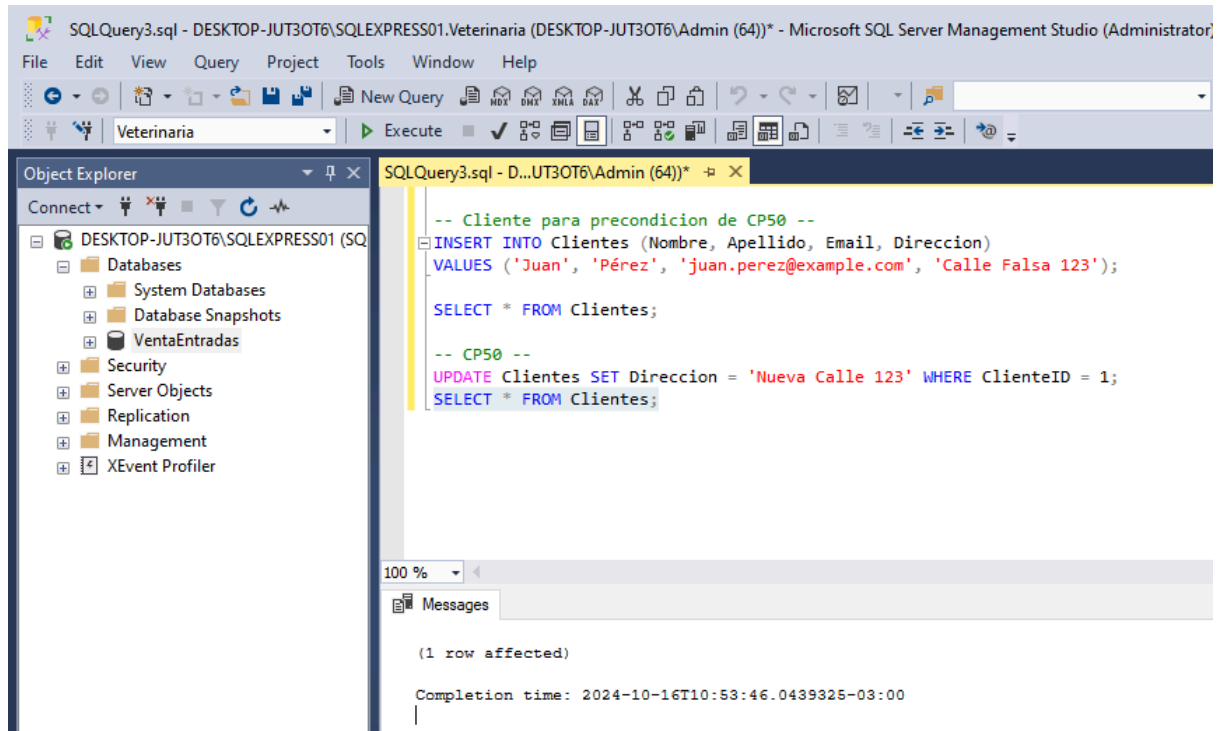
Como se puede observar en la Figura 8, al ejecutar la sentencia en SQL se obtiene la siguiente mensaje: "Violación de la restricción de clave única 'UQ__Clientes__A9D105344F26D771'. No se puede insertar una clave duplicada en el objeto 'dbo.Clientes'. El valor de clave duplicado es (juan.perez@example.com). La instrucción ha sido terminada.;". La inserción falla coincidiendo con el resultado esperado, Se cambia estado a aprobado en tabla de control de casos de prueba.

CP50:

ID	DESCRIPCIÓN	PASOS A SEGUIR	RESULTADOS ESPERADOS
CP50	Actualizar la dirección de un cliente existente en la BBDD.	UPDATE Clientes SET Direccion = 'Nueva Calle 123' WHERE ClienteID = 1;	Se actualiza correctamente la dirección del cliente.

Para este caso de prueba las condiciones son que la tabla Clientes ya esté creada en la BBDD (Figura 2) y que exista un cliente registrado y validado con ClienteID = 1, por Figura 7 y restricción de CP48, ya se encuentran las condiciones cumplidas la BBDD.

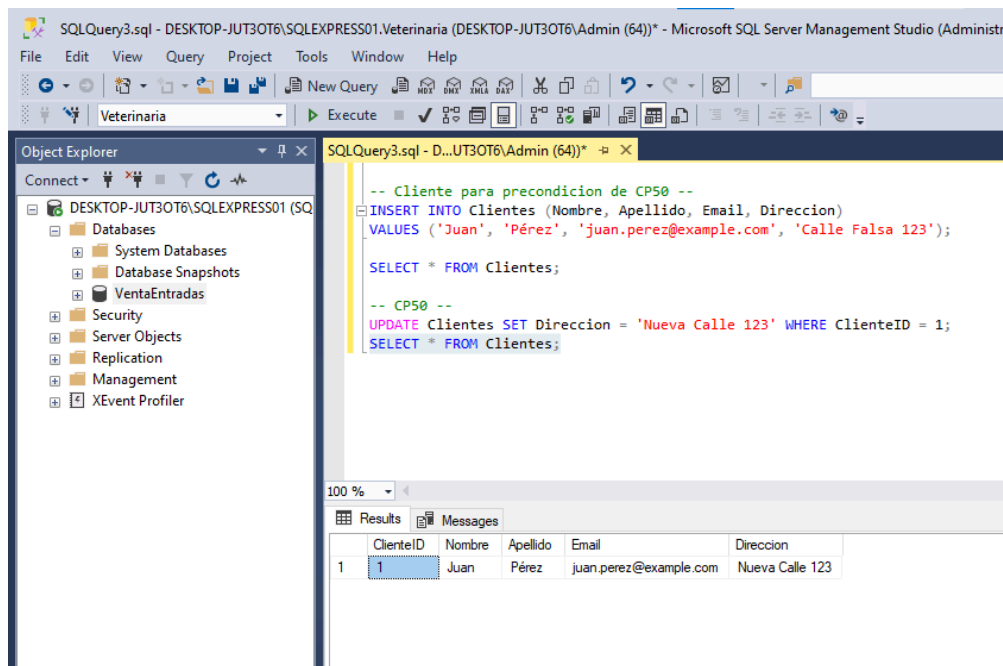
Se procede a insertar la sentencia para verificar el resultado:



(Figura 9 - CP50)

Resultado:

Como se puede observar en la Figura 9 se obtiene: "1 row affected" indicando que la operación realizada afectó a una fila de la tabla. Al verificar la tabla con la sentencia "SELECT * FROM Clientes;" se observa la modificación (Figura 10).



(Figura 10 - Verificación CP50)

CP51:

ID	DESCRIPCIÓN	PASOS A SEGUIR	RESULTADOS ESPERADOS
CP51	Eliminar un cliente que no tiene mascotas asociadas en la base de datos	DELETE FROM Clientes WHERE ClienteID = 2;	El cliente se elimina correctamente.

Para este caso de prueba las precondiciones son que la tabla Clientes ya esté creada en la BBDD (Figura 2) y que exista un cliente registrado y validado con ClienteID = 2, se procede a cargar en la BBDD dicho cliente para que se cumplan las precondiciones:

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. The left pane displays the 'Object Explorer' with the 'Databases' folder expanded, showing 'System Databases', 'Database Snapshots', 'VistaEntradas', 'Security', 'Server Objects', 'Replication', 'Management', and 'XEvent Profiler'. The right pane shows a SQL query in the 'SQLQuery1.sql' window. The query includes a CREATE TABLE statement for 'Usuarios', two INSERT statements for 'Clientes' (one for ID 1 and one for ID 2), and a SELECT statement to verify the contents of the 'Clientes' table. Below the query, the 'Results' pane displays the output of the SELECT statement, showing two rows of data from the 'Clientes' table.

```

CREATE TABLE Usuarios (
    UsuarioID INT IDENTITY(1,1) PRIMARY KEY,
    NombreUsuario NVARCHAR(50) NOT NULL UNIQUE,
    Contraseña NVARCHAR(255) NOT NULL,
    Rol NVARCHAR(50) NOT NULL CHECK (Rol IN ('Administrador', 'Empleado'))
);

-- Cliente ID = 1 --
INSERT INTO Clientes (Nombre, Apellido, Email, Direccion)
VALUES ('Juan', 'Pérez', 'juan.perez@example.com', 'Calle Falsa 123');

-- ClienteID = 2 --
INSERT INTO Clientes (Nombre, Apellido, Email, Direccion)
VALUES ('María', 'Gómez', 'maria.gomez@example.com', 'Calle 456');

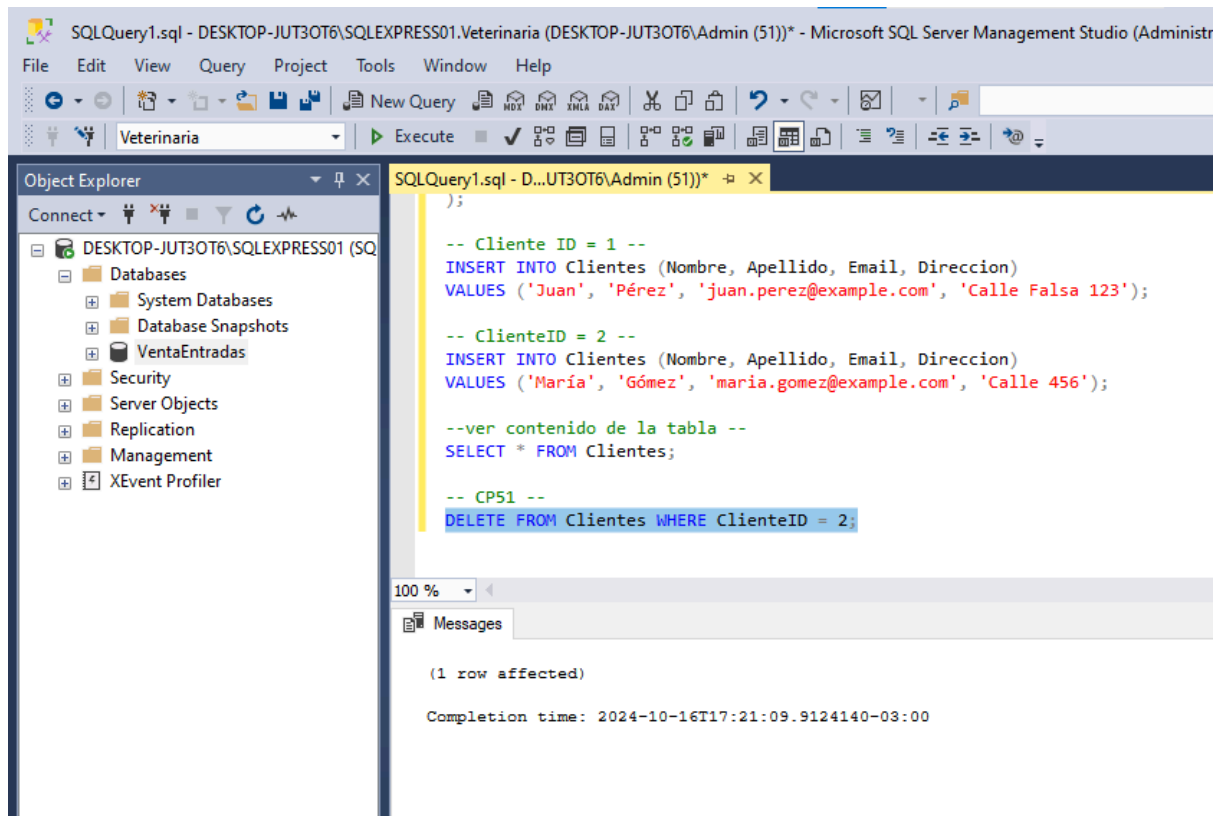
--ver contenido de la tabla --
SELECT * FROM Clientes;

```

ClienteID	Nombre	Apellido	Email	Direccion
1	Juan	Pérez	juan.perez@example.com	Calle Falsa 123
2	María	Gómez	maria.gomez@example.com	Calle 456

(Figura 11 - Carga de cliente para CP51)

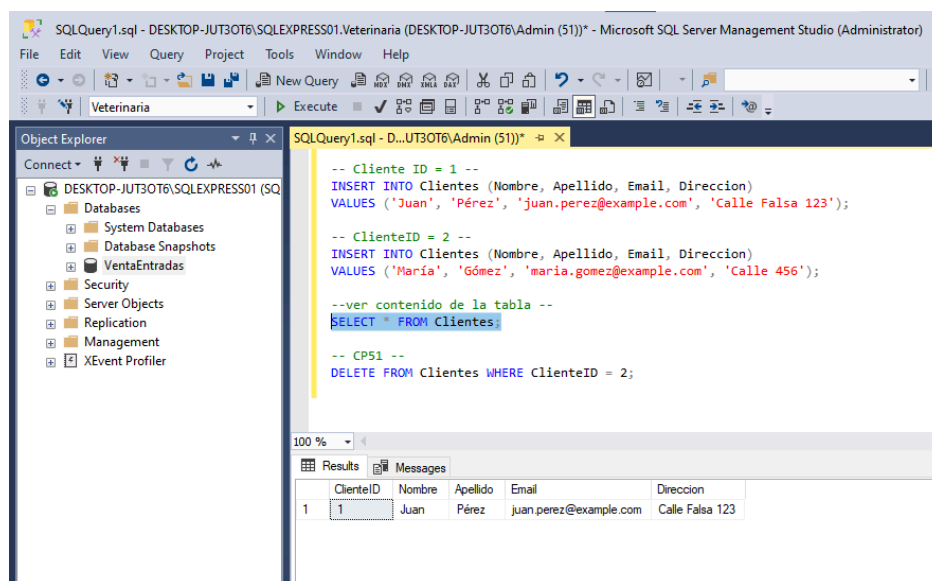
Se procede a insertar la sentencia para verificar el resultado:



(Figura 12 - CP51)

Resultado:

Como se puede observar en la Figura 12 se obtiene: "1 row affected" indicando que la operación realizada afectó a una fila de la tabla. Al verificar la tabla con la sentencia "SELECT * FROM Clientes;" se observa la modificación (Figura 13).

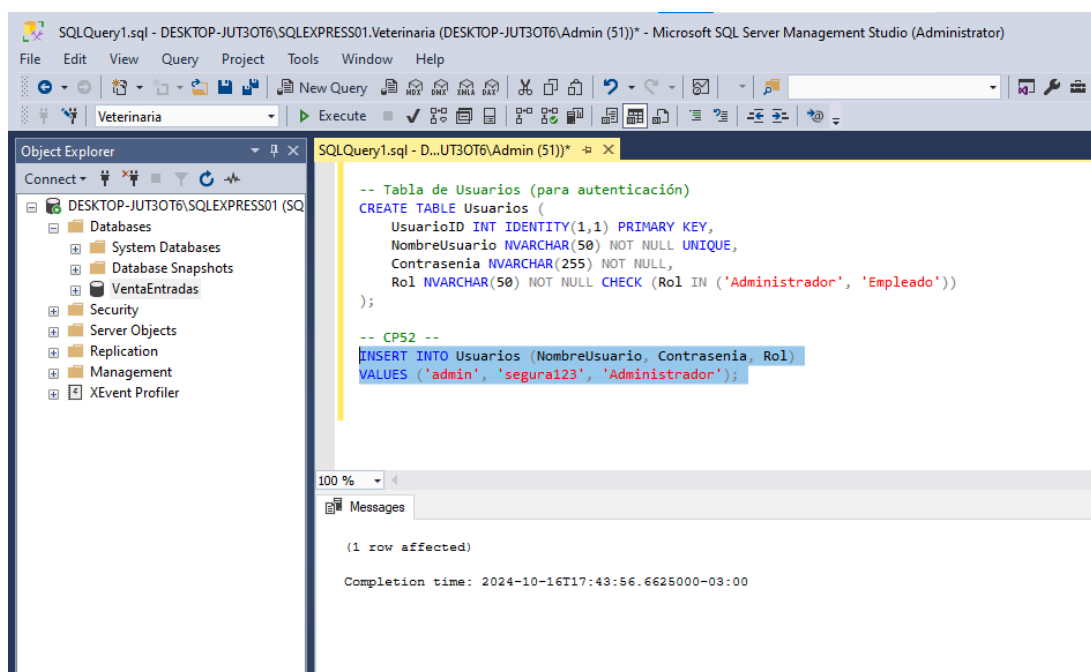


(Figura 13 - verificación CP51)

CP52:

ID	DESCRIPCIÓN	PASOS A SEGUIR	RESULTADOS ESPERADOS
CP52	Inserción de usuario con todos los datos válidos.	INSERT INTO Usuarios (NombreUsuario, Contraseña, Rol) VALUES ('admin', 'segura123', 'Administrador');	Inserción exitosa, se genera UsuarioID.

Para este caso de prueba las precondiciones son que la tabla Usuarios ya esté creada en la BBDD (Figura 5). Se procede a insertar la sentencia para observar el resultado:



The screenshot displays the Microsoft SQL Server Management Studio interface. The title bar indicates the file 'SQLQuery1.sql' is open in the 'Veterinaria' database. The Object Explorer on the left shows the database structure, including 'Databases', 'System Databases', 'Database Snapshots', 'VentasEntradas', 'Security', 'Server Objects', 'Replication', 'Management', and 'XEvent Profiler'. The main query window shows the following SQL code:

```
-- Tabla de Usuarios (para autenticación)
CREATE TABLE Usuarios (
    UsuarioID INT IDENTITY(1,1) PRIMARY KEY,
    NombreUsuario NVARCHAR(50) NOT NULL UNIQUE,
    Contraseña NVARCHAR(255) NOT NULL,
    Rol NVARCHAR(50) NOT NULL CHECK (Rol IN ('Administrador', 'Empleado'))
);

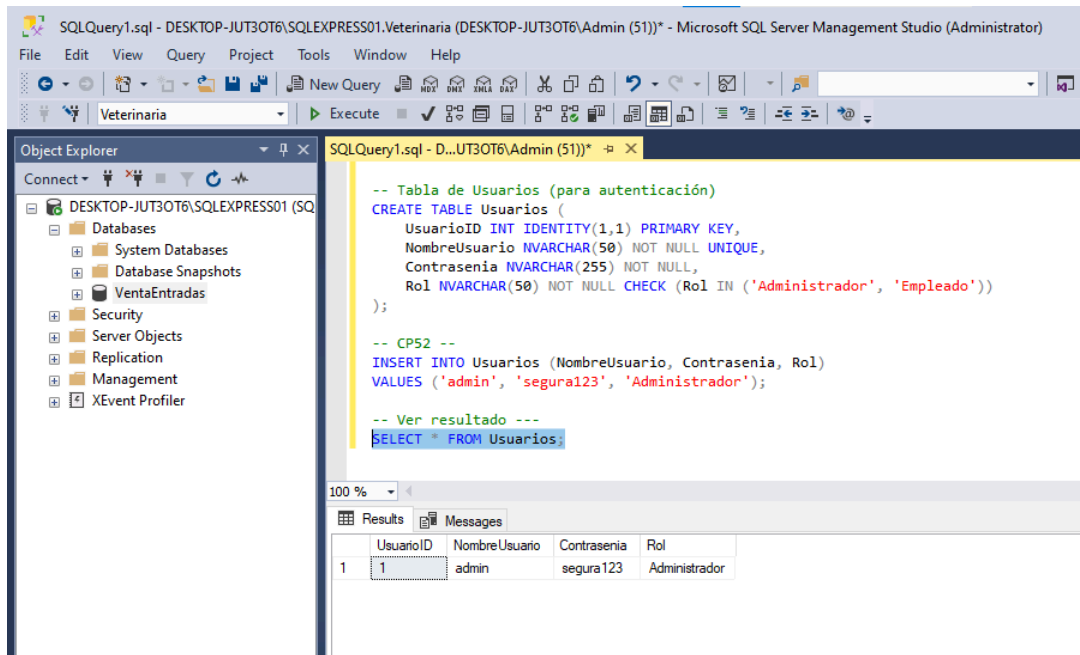
-- CP52 --
INSERT INTO Usuarios (NombreUsuario, Contraseña, Rol)
VALUES ('admin', 'segura123', 'Administrador');
```

The Messages pane at the bottom shows the execution result: "(1 row affected)" and "Completion time: 2024-10-16T17:43:56.6625000-03:00".

(Figura 14 - CP52)

Resultado:

Como se puede observar en la Figura 14 se obtiene: "1 row affected" indicando que la operación realizada afectó a una fila de la tabla. Al verificar la tabla con la sentencia "SELECT * FROM Usuarios;" se observa la modificación (Figura 15).

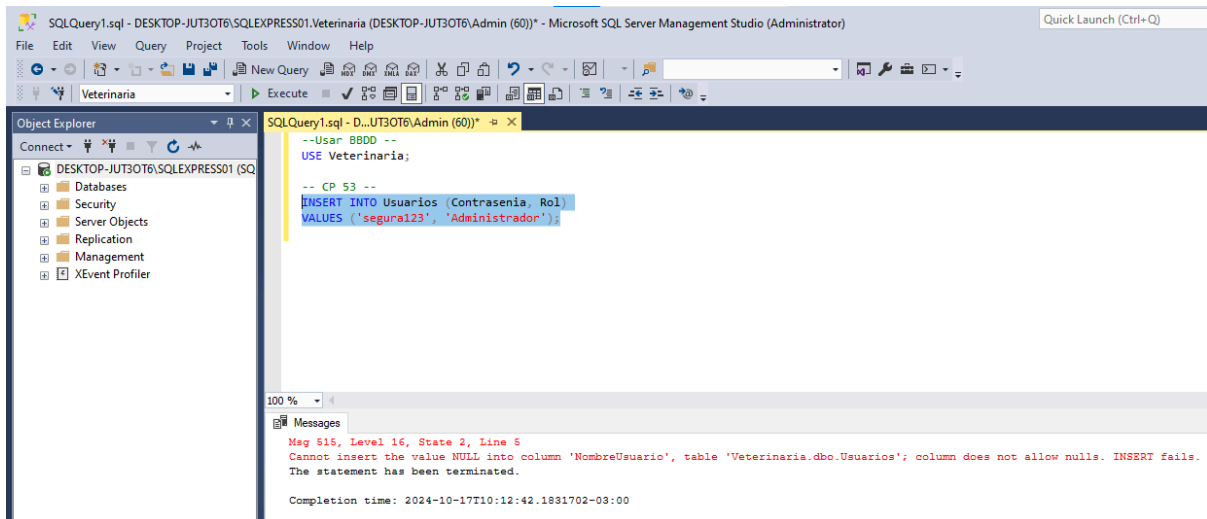


(Figura 15 - Verificación de CP52)

CP53:

ID	DESCRIPCIÓN	PASOS A SEGUIR	RESULTADOS ESPERADOS
CP53	Intentar insertar usuario sin el campo NombreUsuario	INSERT INTO Usuarios (Contraseña, Rol) VALUES ('segura123', 'Administrador');	Error de restricción NOT NULL.

Para este caso de prueba las precondiciones son que la tabla Usuarios ya esté creada en la BBDD (Figura 5). Se procede a insertar la sentencia para observar el resultado:



(Figura 16 - CP53)

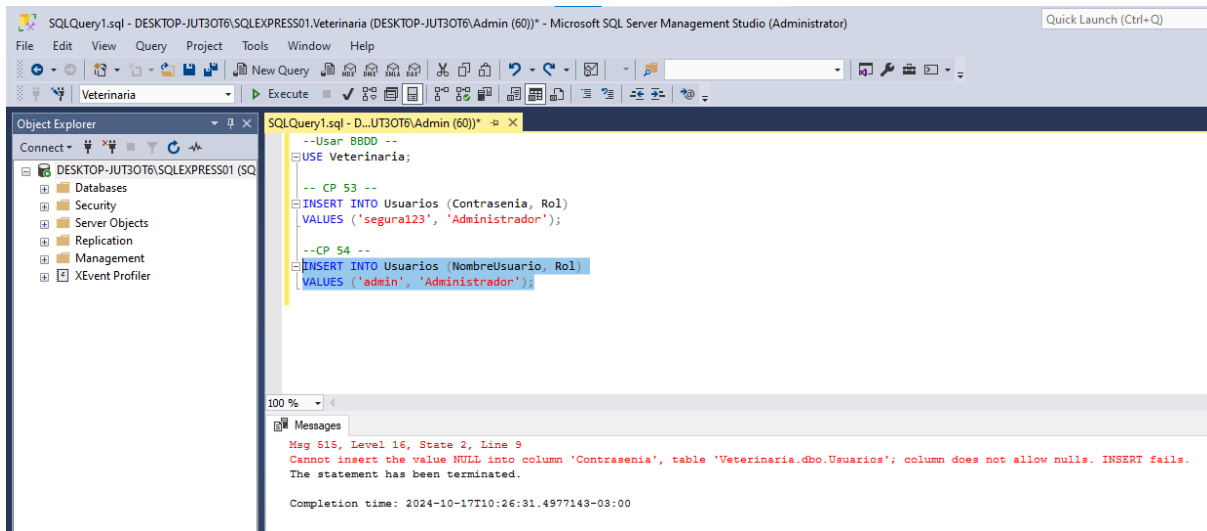
Resultado:

Como se puede observar en la Figura 16, se obtiene el mensaje: *"Msg 515, Level 16, State 2, Line 5: Cannot insert the value NULL into column 'NombreUsuario', table 'Veterinaria.dbo.Usuarios'; column does not allow nulls. INSERT fails. The statement has been terminated."* Esto indica que la operación de inserción ha fallado debido a que se intentó insertar un valor NULL en la columna NombreUsuario, la cual no permite valores nulos.

CP54:

ID	DESCRIPCIÓN	PASOS A SEGUIR	RESULTADOS ESPERADOS
CP54	Intentar insertar usuario sin el campo Contraseña	INSERT INTO Usuarios (NombreUsuario, Rol) VALUES ('admin', 'Administrador');	Error de restricción NOT NULL.

Para este caso de prueba las precondiciones son que la tabla Usuarios ya esté creada en la BBDD (Figura 5). Se procede a insertar la sentencia para observar el resultado:



(Figura 17 - CP54)

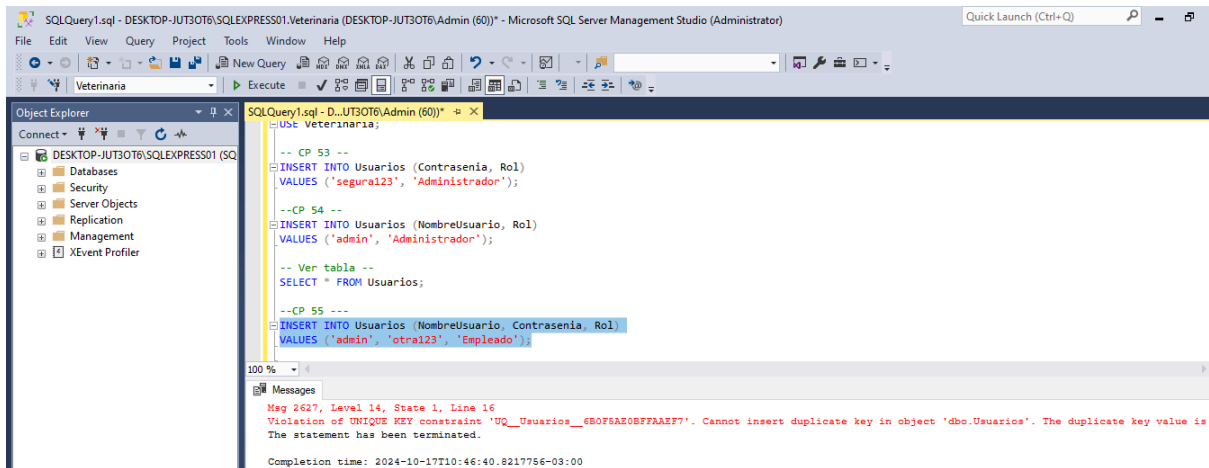
Resultado:

Como se puede observar en la Figura 17, se obtiene el mensaje: *"Msg 515, Level 16, State 2, Line 9: Cannot insert the value NULL into column 'Contrasenia', table 'Veterinaria.dbo.Usuarios'; column does not allow nulls. INSERT fails. The statement has been terminated."* Esto indica que la operación de inserción ha fallado debido a que se intentó insertar un valor NULL en la columna Contraseña, la cual no permite valores nulos.

CP55:

ID	DESCRIPCIÓN	PASOS A SEGUIR	RESULTADOS ESPERADOS
CP55	Intentar insertar un usuario con un nombre de usuario que ya existe en la BBDD	INSERT INTO Usuarios (NombreUsuario, Contraseña, Rol) VALUES ('admin', 'otra123', 'Empleado');	Error de restricción UNIQUE.

Para este caso de prueba las condiciones son que la tabla Usuarios ya esté creada en la BBDD (Figura 5), y que exista un Usuario registrado y validado con el NombreUsuario que intentamos validar (Figura 15). Se procede a insertar la sentencia para observar el resultado:



(Figura 18 - CP55)

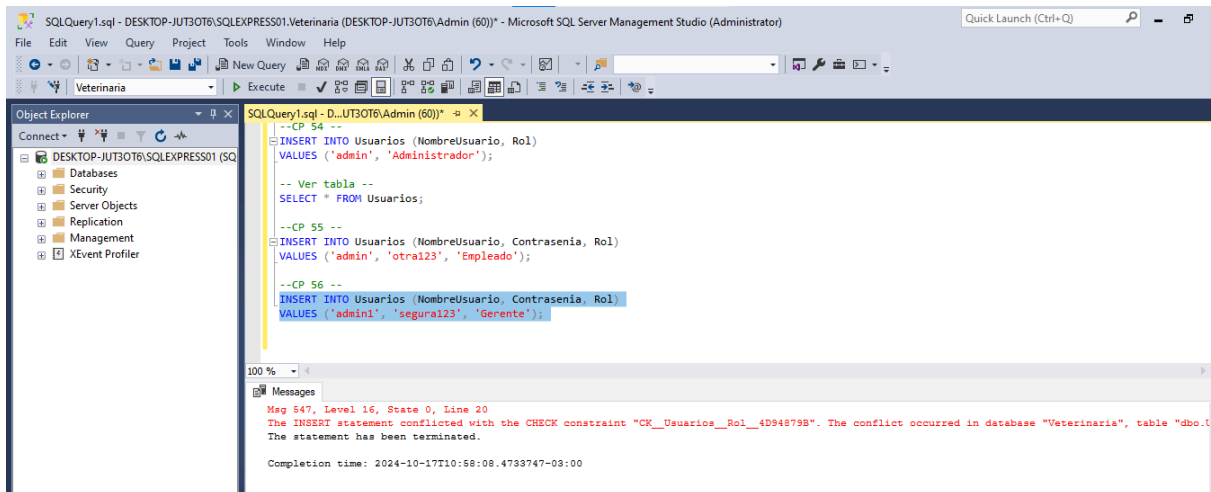
Resultado:

Como se puede observar en la Figura 18, se obtiene el mensaje: *"Msg 2627, Level 14, State 1, Line 16: Violation of UNIQUE KEY constraint 'UQ_Usuarios__6B0F5AE0BFFAAEF7'. Cannot insert duplicate key in object 'dbo.Usuarios'. The duplicate key value is (admin). The statement has been terminated."* Esto indica que la operación de inserción ha fallado debido a que se intentó insertar un valor duplicado en la columna que tiene una restricción de clave única. En este caso, el valor duplicado es (admin).

CP56:

ID	DESCRIPCIÓN	PASOS A SEGUIR	RESULTADOS ESPERADOS
CP56	Intentar insertar usuario con un rol no permitido	INSERT INTO Usuarios (NombreUsuario, Contraseña, Rol) VALUES ('admin1', 'segura123', 'Gerente');	Error de restricción CHECK.

Para este caso de prueba las precondiciones son que la tabla Usuarios ya esté creada en la BBDD (Figura 5). Se procede a insertar la sentencia para observar el resultado:



(Figura 19 - CP56)

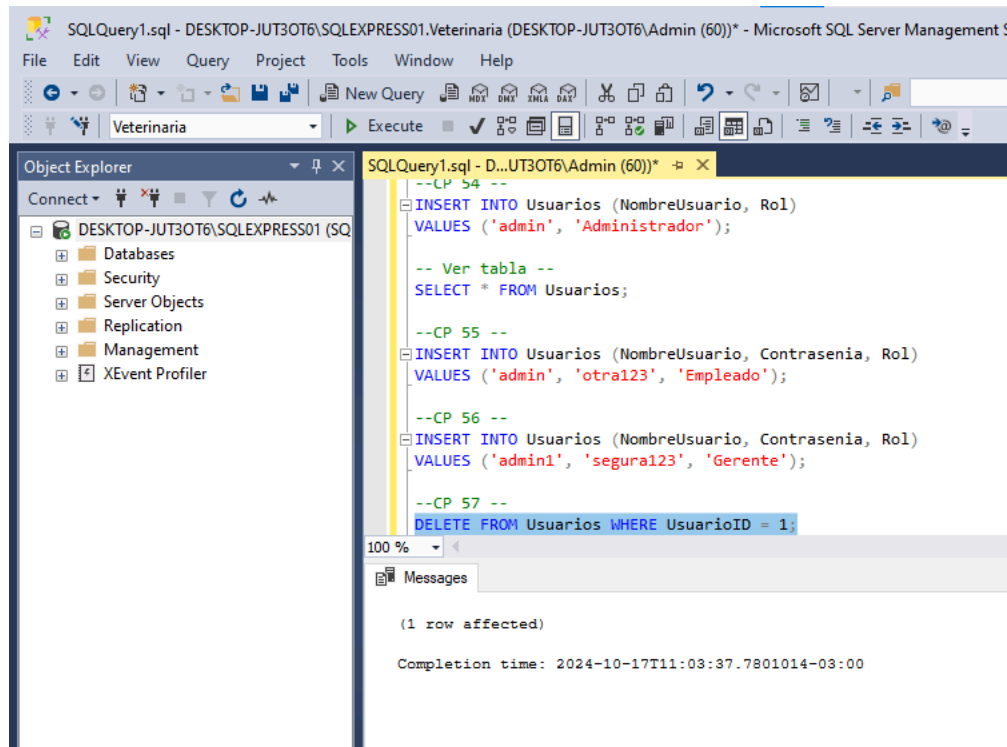
Resultado:

Como se puede observar en la Figura 19, se obtiene el mensaje: "Msg 547, Level 16, State 0, Line 20: The INSERT statement conflicted with the CHECK constraint 'CK__Usuarios__Rol__4D94879B'. The conflict occurred in database 'Veterinaria', table 'dbo.Usuarios', column 'Rol'. The statement has been terminated." Esto indica que la operación de inserción ha fallado debido a que el valor insertado en la columna Rol no cumple con la restricción definida por la cláusula CHECK.

CP57:

ID	DESCRIPCIÓN	PASOS A SEGUIR	RESULTADOS ESPERADOS
CP57	Eliminar un usuario existente de la BBDD	DELETE FROM Usuarios WHERE UsuarioID = 1;	El usuario se elimina correctamente de la tabla Usuarios.

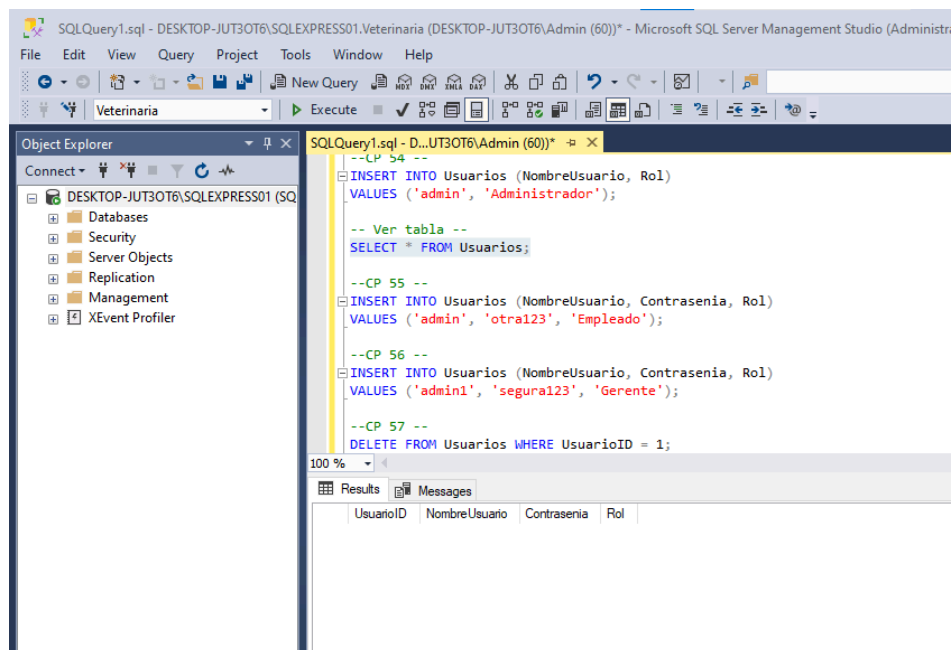
Para este caso de prueba las precondiciones son que la tabla Usuarios ya esté creada en la BBDD (Figura 5), y que exista un Usuario registrado y validado con el NombreUsuario que intentamos validar (Figura 15). Se procede a insertar la sentencia para observar el resultado:



(Figura 20 - CP57)

Resultado:

Como se puede observar en la Figura 20 se obtiene: "1 row affected" indicando que la operación realizada afectó a una fila de la tabla. Al verificar la tabla con la sentencia "SELECT * FROM Usuarios," se observa la modificación (Figura 21).

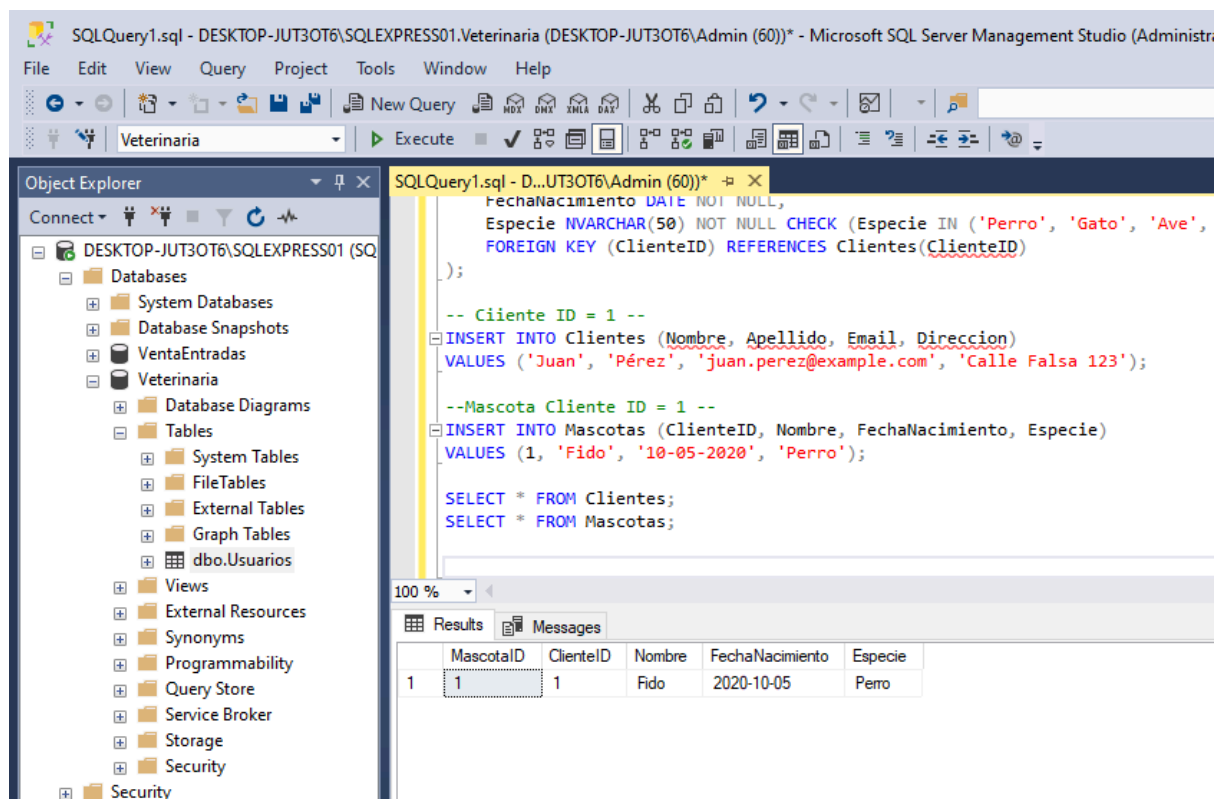


(Figura 21 - Verificación de CP57)

CP58:

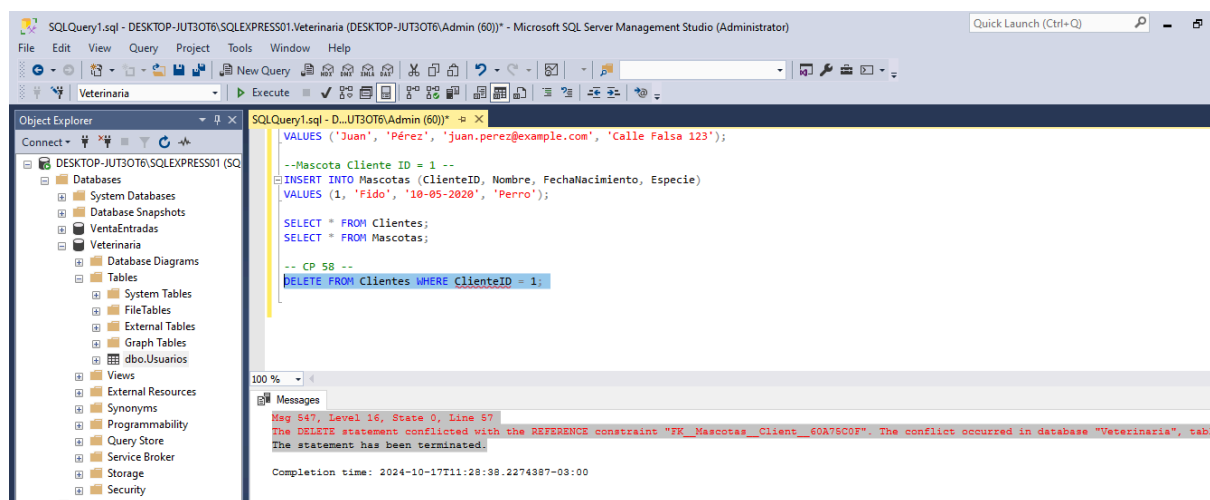
ID	DESCRIPCIÓN	PASOS A SEGUIR	RESULTADOS ESPERADOS
CP58	Intentar eliminar un cliente que tiene mascotas registradas.	DELETE FROM Clientes WHERE ClienteID = 1;	Error de restricción FOREIGN KEY (no se permite eliminar).

Para este caso de prueba las precondiciones son que la tabla Clientes y la tabla Mascotas ya estén creadas en la BBDD (Figura 2 y 3). que exista un Cliente registrado y validado con el ClienteID requerido para este caso, y una mascota asociada al mismo como se observa en la Figura 22:



(Figura 22 - Verificación de precondiciones CP58)

Se procede a insertar la sentencia para observar el resultado:



(Figura 23 - CP58)

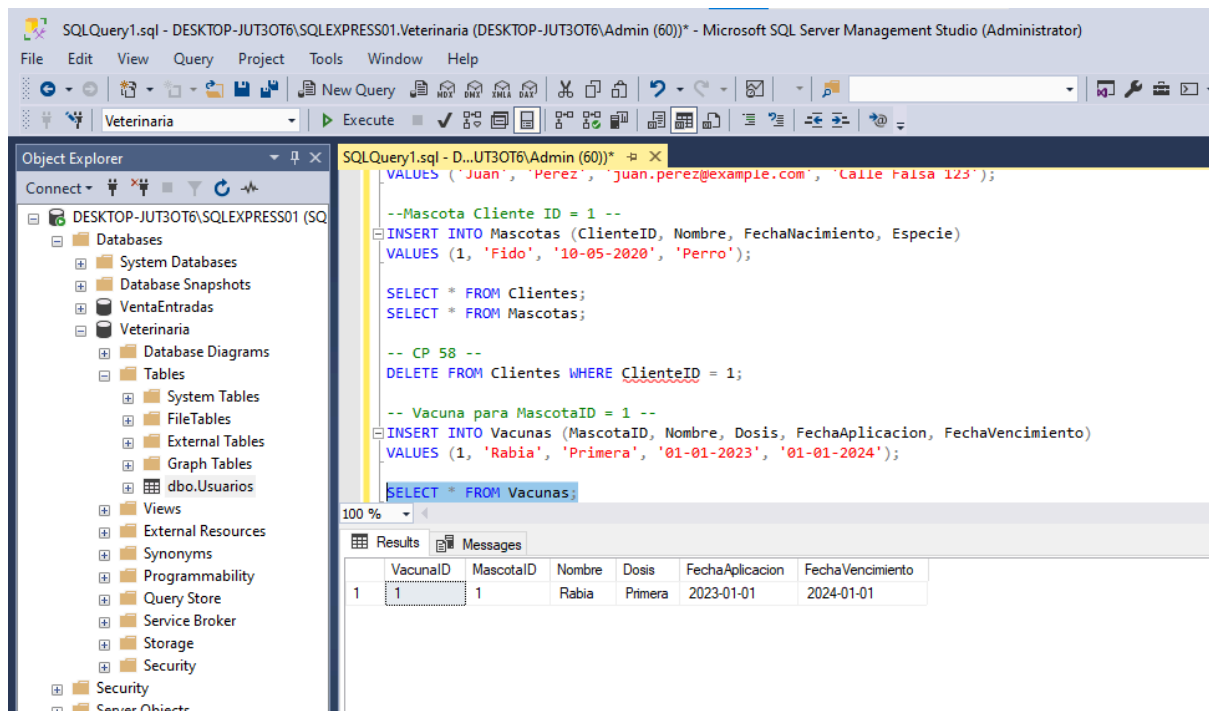
Resultado:

Como se puede observar en la Figura 23, se obtiene el mensaje: *"Msg 547, Level 16, State 0, Line 57: The DELETE statement conflicted with the REFERENCE constraint 'FK_Mascotas_Client_60A75C0F'. The conflict occurred in database 'Veterinaria', table 'dbo.Mascotas', column 'ClienteID'. The statement has been terminated."* Esto indica que la operación de eliminación ha fallado porque se intentó eliminar un registro de la tabla Clientes que está siendo referenciado por la columna ClienteID en la tabla Mascotas. Es decir, no se puede eliminar un cliente que todavía tiene registros asociados en la tabla Mascotas.

CP59:

ID	DESCRIPCIÓN	PASOS A SEGUIR	RESULTADOS ESPERADOS
CP59	Intentar eliminar una mascota que tiene vacunas registradas.	DELETE FROM Mascotas WHERE MascotaID = 1;	Error de restricción FOREIGN KEY (no se permite eliminar).

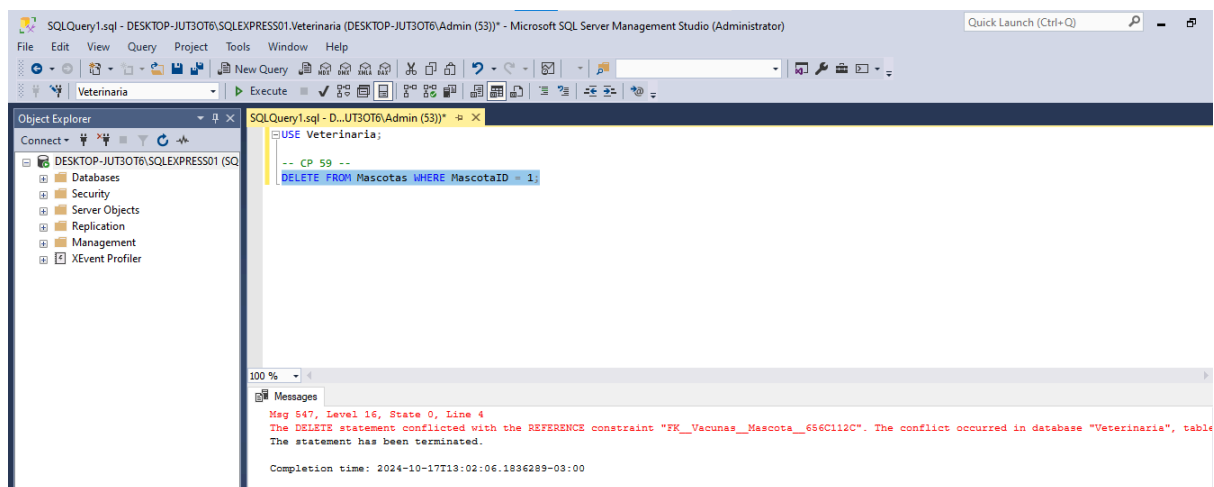
Para este caso de prueba las precondiciones son que la tabla Mascotas y la tabla Vacunas ya estén creadas en la BBDD (Figura 3 y 4). que exista una Vacuna registrada y validada asociada a el MascotaID requerido para este caso, como se observa en la Figura 24:



(Figura 24 - Verificación de precondiciones para CP59)

Resultado:

Como se puede observar en el mensaje de error, "Msg 547, Level 16, State 0, Line 4: The DELETE statement conflicted with the REFERENCE constraint 'FK__Vacunas__Mascota__656C112C'. The conflict occurred in database 'Veterinaria', table 'dbo.Vacunas', column 'MascotaID'." Esto indica que la operación de eliminación ha fallado porque se intentó eliminar un registro de la tabla Mascotas que está siendo referenciado por la columna MascotaID en la tabla Vacunas. En otras palabras, no se puede eliminar una mascota que todavía tiene registros asociados en la tabla Vacunas.

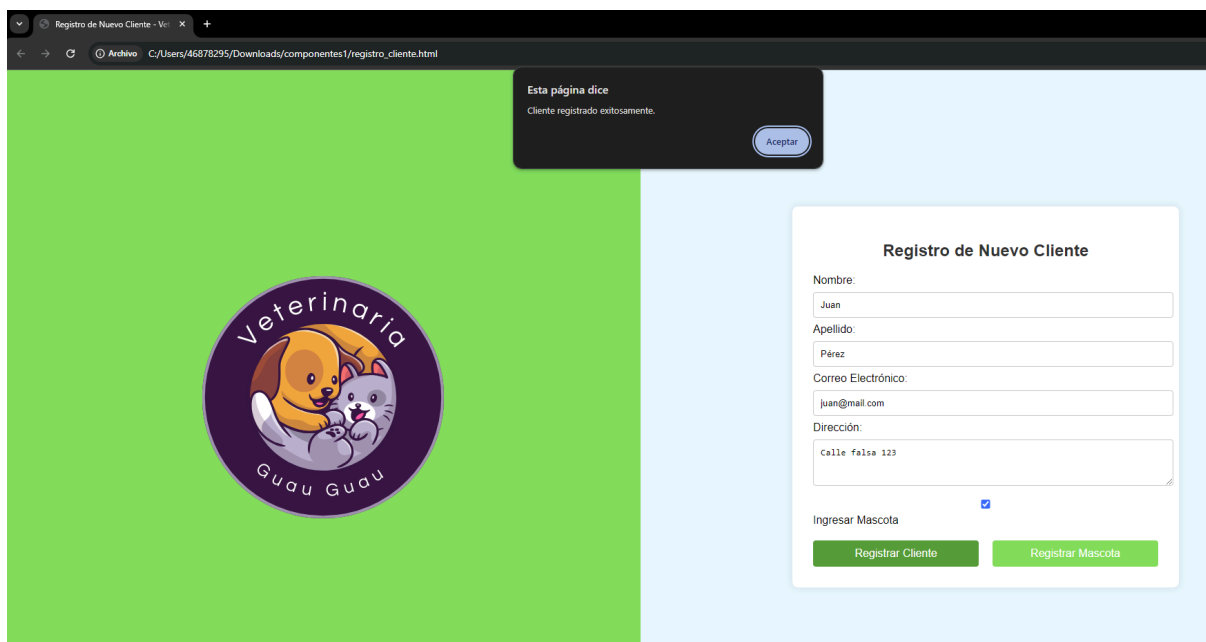


(Figura 23 - CP59)

CP61:

ID	DESCRIPCIÓN	PASOS A SEGUIR	RESULTADOS ESPERADOS
CP61	Registrar un cliente seleccionando botón de ingresar la mascota, luego de validados los datos del cliente pasa a ingresar mascota.	<ol style="list-style-type: none">1. Acceder a la página de registro.2. Ingresar "Juan" en el campo Nombre.3. Ingresar "Pérez" en el campo Apellido.4. Ingresar "juan@mail.com" en el campo Email.5. Ingresar "Calle Falsa 123" en el campo Dirección.6. Hacer clic en el botón de Ingresar Mascota.7. Hacer clic en botón aceptar de ventana emergente de registro exitoso8. Se abre registro de mascotas	Registro exitoso de cliente, mensaje de confirmación, apertura de página de Registro de mascotas.

Para este caso de prueba las precondiciones Cliente no registrado en el sistema. Se ejecuta el mismo obteniendo los siguientes resultados:



(Figura 24 - CP61 en registro_clientes.html)

Resultado:

El resultado obtenido en este caso de prueba sugiere un problema relacionado con la gestión de archivos, lo que podría estar afectando el flujo de registro de la mascota. El mensaje de error **"err_file_not_found"** indica que el sistema no puede acceder a un archivo necesario para completar la operación (Figura 25).



No se ha podido acceder al archivo

Es posible que se haya movido, editado o eliminado.

ERR_FILE_NOT_FOUND

(Figura 25 - Resultado CP61)

Por otra parte, se entiende que la inconsistencia surge en la configuración de la dirección, si el sistema está buscando el archivo en una ubicación incorrecta, se debe verificar cómo está definida la ruta en el código.

Se observa que en línea 128 del código, la dirección es hacia login.html cuando debería ser a registro_mascota (Figura 26).

```
<script>
  document.getElementById('clienteForm').addEventListener('submit', function(e) {
    e.preventDefault();
    // Validación simple
    const nombre = document.getElementById('nombre').value.trim();
    const apellido = document.getElementById('apellido').value.trim();
    const email = document.getElementById('email').value.trim();
    const direccion = document.getElementById('direccion').value.trim();
    if(nombre && apellido && email && direccion){
      alert("Cliente registrado exitosamente.");
      window.location.href = "login.html"; /* aquí debería direccionar a registro_mascota no a login.html */
    } else {
      document.getElementById('error').innerText = "Por favor, complete todos los campos.";
    }
  });
});
```

(Figura 26 - código observado en bloc de notas)

Se crea incidente en herramienta de gestión Jira para el reporte del mismo, con código: PT2-137.

Conclusión:

El alcance de este informe se ha enfocado en evaluar las pruebas realizadas para verificar la correcta implementación de las tablas y sus restricciones asociadas, asegurando que todos los datos se almacenen conforme a las reglas definidas por el cliente.

Integridad de Datos: Las pruebas realizadas han demostrado que las restricciones implementadas en la base de datos son efectivas para garantizar la integridad de la información. Se verificó con éxito la correcta aplicación de restricciones como NOT NULL, UNIQUE y FOREIGN KEY, asegurando que las operaciones de inserción, actualización y eliminación se ajusten a las normas establecidas.

Gestión de Errores: La identificación de mensajes de error específicos durante las pruebas indica que el sistema está bien diseñado para manejar adecuadamente las violaciones de restricciones. Esto no solo mejora la experiencia del usuario, sino que también previene la corrupción de datos.

Independencia de la Tabla Usuarios: La tabla Usuarios se ha mostrado completamente independiente de las demás tablas del sistema. Al no contener claves foráneas ni ser referenciada por otras tablas, su existencia y funcionamiento no dependen de los datos en las tablas de Clientes, Mascotas o Vacunas. Esta autonomía permite una gestión eficiente y segura de la autenticación y los roles de usuario, garantizando la integridad de la información relacionada con clientes y mascotas.

Consistencia en las Relaciones: Las pruebas de consistencia confirmaron que las relaciones entre las tablas se mantienen intactas. Las restricciones FOREIGN KEY evitaron la eliminación de registros en tablas referenciadas, asegurando así la integridad referencial de los datos.

Por otro lado se tuvo como objetivo verificar el correcto flujo de registro de un cliente y la posterior redirección a la página de registro de mascotas. Sin embargo, el resultado obtenido reveló una falla en la gestión de archivos, lo que impidió completar el proceso correctamente. El error **"err_file_not_found"** indicó que el sistema no pudo acceder a un archivo necesario para el flujo, lo que sugiere un problema en la configuración de las rutas.

En resumen, la base de datos de la veterinaria Guau Guau ha sido validado de manera efectiva en cuanto a la integridad y consistencia de los datos, lo que garantiza un manejo eficiente de la información de clientes y mascotas. Sin embargo, es importante señalar que, si bien este informe valida ciertos aspectos del sistema, se requiere una evaluación completa de los demás casos e instancias de la BBDD para poder ofrecer una validación total.

Un aspecto crucial de este sistema es la tabla Usuarios, que opera de forma completamente autónoma. Al no contener claves foráneas ni ser referenciada por otras tablas, su existencia y funcionamiento no dependen de los datos en las tablas de Clientes, Mascotas o Vacunas. Esta independencia permite una gestión eficaz y segura de la autenticación y los roles de usuario, asegurando que la integridad de la información relacionada con clientes y mascotas no se vea comprometida.

En el caso de la versión del software web se observan incidentes relacionados con la gestión de archivos que impide el flujo normal y que las acciones sean completadas satisfactoriamente, se ofrece una alternativa en la Figura 26, de posible solución.

Anexos:

1. **Planilla de casos de prueba** [Acceder a planilla.](#)
2. **Proyecto en Jira:** [Acceder al proyecto](#)