**Manual de usuario de los resultados de la prueba para Analista Desarrollador de Base de Datos.**

 Edwin Dominguez

Salvador Cuevas

La primera situación planteada a realizar es la siguiente:

1. Favor autogenerar un secuencial para productos de almacenes de MARDOM, se le debe pasar el producto y el tipo de producto y con esto debe generar un codigo secuencial único e irrepetible. **Ejemplo**: MAP-Seco-Ropa-Feb-23-2021.

a. Se debe crear una entidad tipo productos con los siguientes datos:

i. Seco

ii. Frio

iii. Peligroso

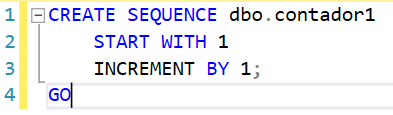
b. Se debe crear una entidad producto con los siguientes datos:

i. Ropa

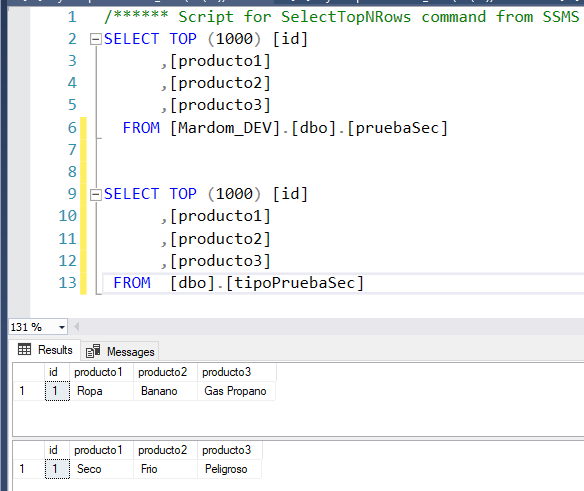
ii. Banano

iii. Gas propano

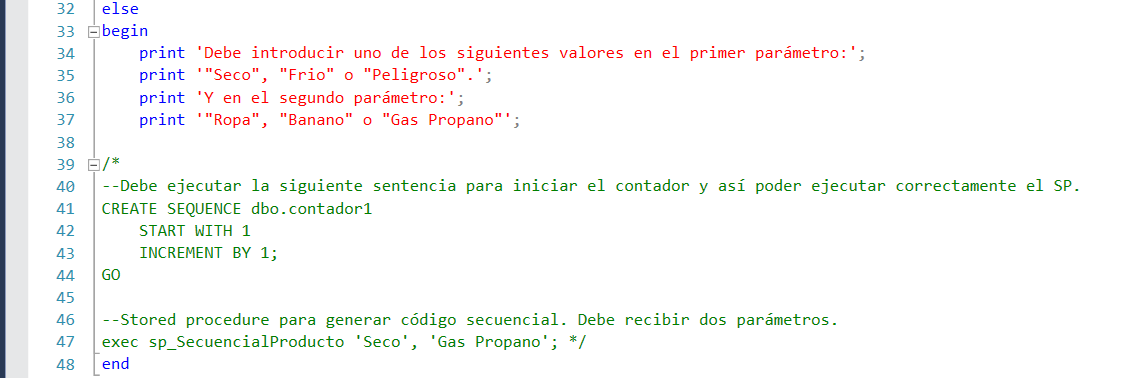
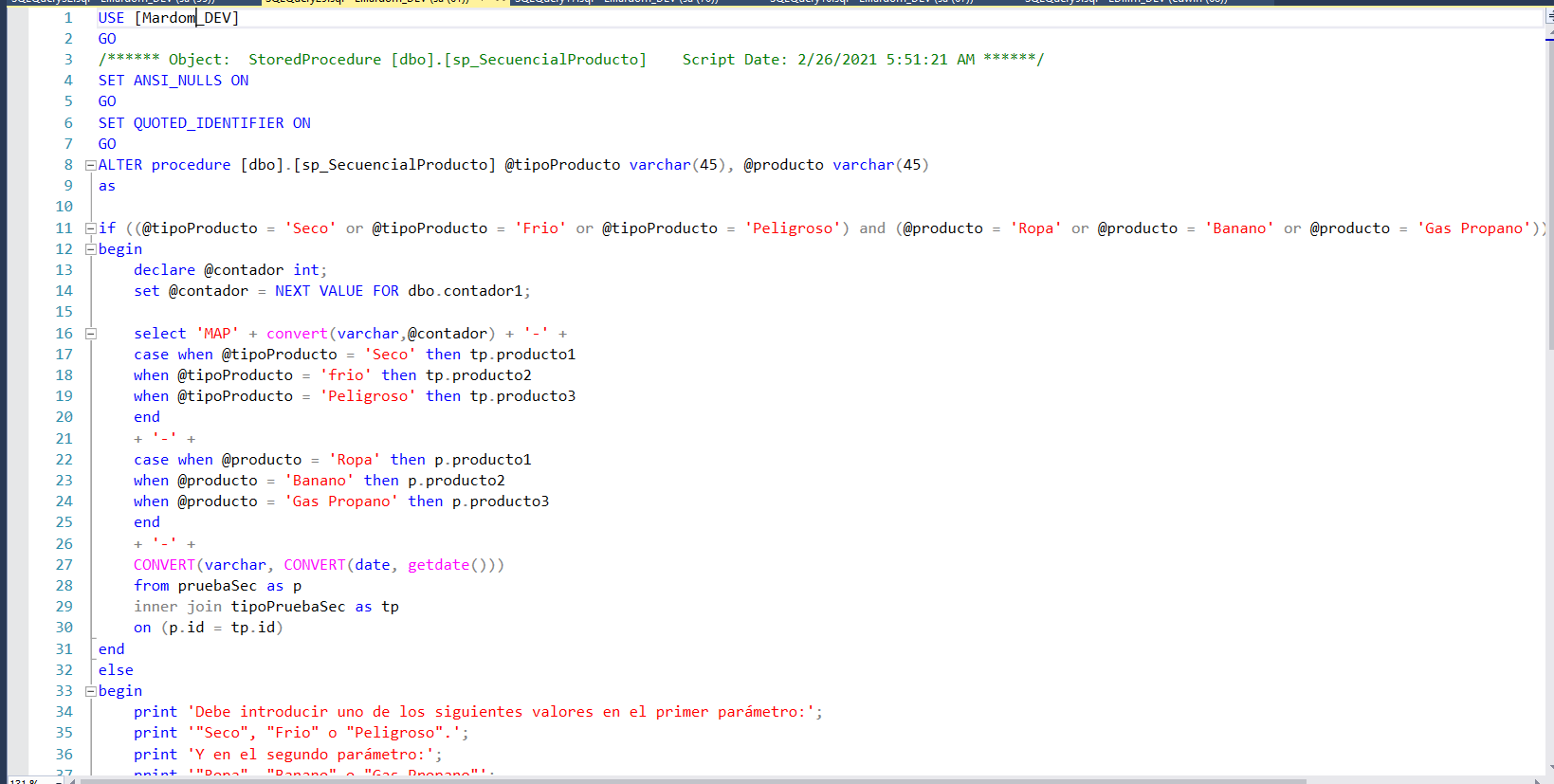
Para la solución a este mandato, primero ejecuté el siguiente query para generar un número que es autoincremental llamado *contador1*:



Luego las siguientes tablas con los valores para las dos entidades requeridas:

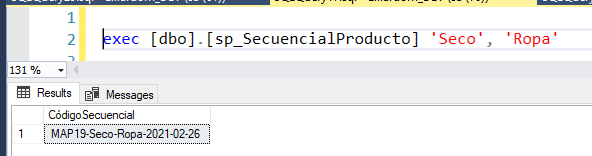


Entonces el siguiente procedimiento almacenado para poder generar la secuencia en base a los datos de las pasadas dos tablas:



sp\_SecuencialProducto debe recibir dos argumentos en sus parámetros, el primero es para el tipo de producto en los cuales están disponibles *Seco*, *Frio* y *Peligroso*. Y como segundo argumento pasarle el producto, los cuales son *Ropa*, *Banano* y *Gas Propano.*

Si por ejemplo, se ejecuta el SP sp\_SecuencialProducto con los siguientes argumentos *‘Seco’* y *‘Ropa’,* generaría el siguiente resultado:



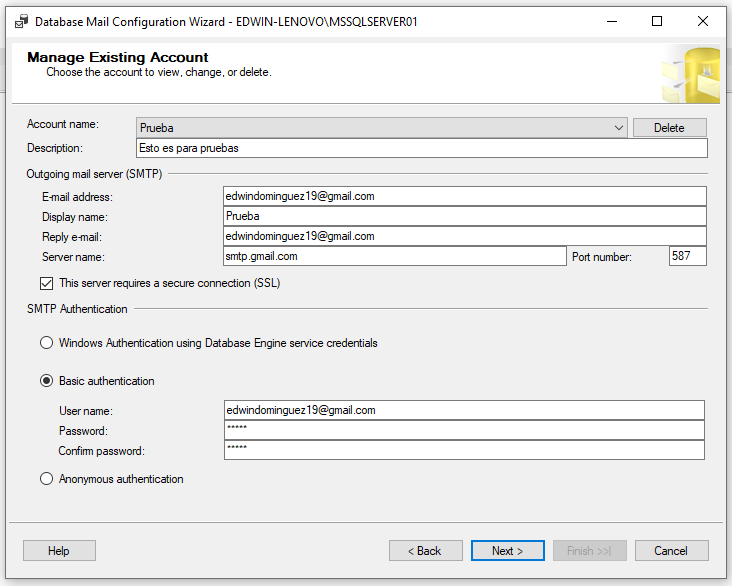
La sentencia NEXT VALUE FOR dbo.contador1 incrementa su valor uno más por cada llamada a él, éste está localizado de lado al MAP y logra que la secuencia nunca se logre repetir.

La segunda situación planteada a realizar es la siguiente:

2. Realizar una estructura para el envió de notificaciones a un correo o varios que se indiquen como parámetros. Los parámetros serian (emails, notificación id)

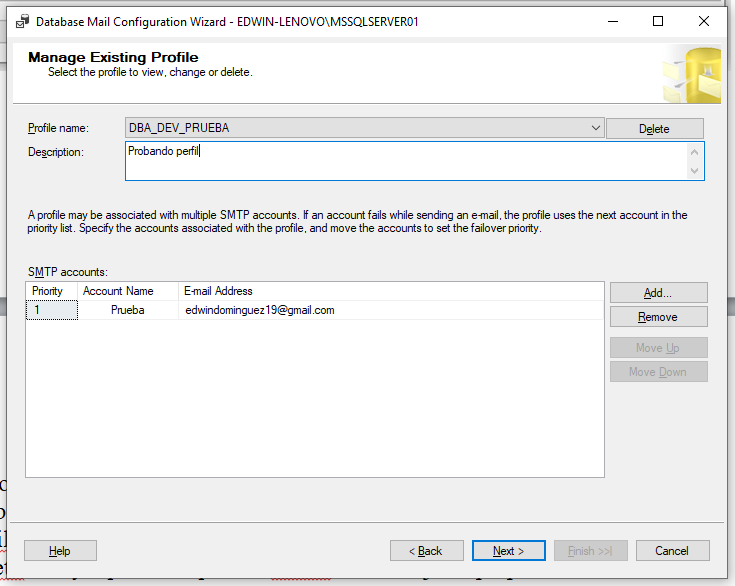
* 1. a. Se debe crear una entidad donde se guarde el texto a enviar. El texto debe decir: “Hola este es un correo de prueba para los destinatarios: {emails}”
  2. b. Se debe guardar un histórico de los envíos de emails, exitosos o no.

Para esto debí hacer una configuración en Database Mail:

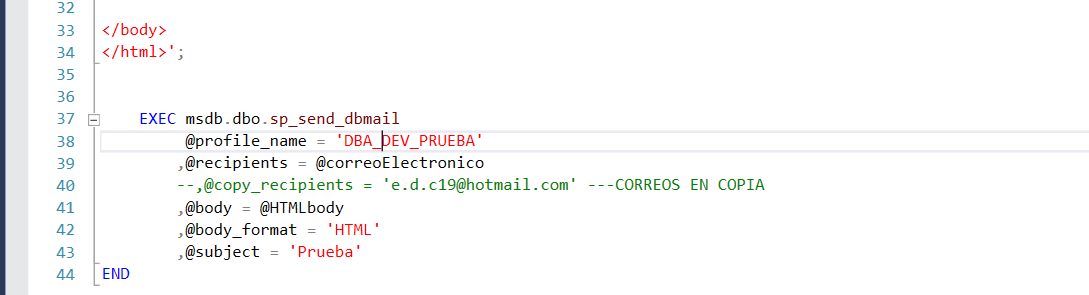


Como se puede notar en la imagen que está justo arriba, se le dio un nombre Prueba a la cuenta de correo en Outgoing mail server elegí una cuenta de Gmail, por lo tanto en Server name usé smtp.gmail.com con el número de puereto 587 ya que es el que usa Gmail. Ya debajo se preparó la autenticación con el mismo correo electrónico.

Luego creé y utllicé el siguite nombre de perfil:

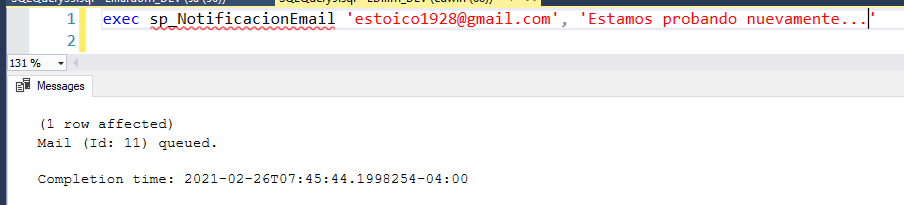


Ya con todo esto configurado, creé un procedimiento almacenado para poder enviar el correo de una manera practica con los parámetros requeridos y a la vez poder almacenar los resultados del procedimiento en un histórico:

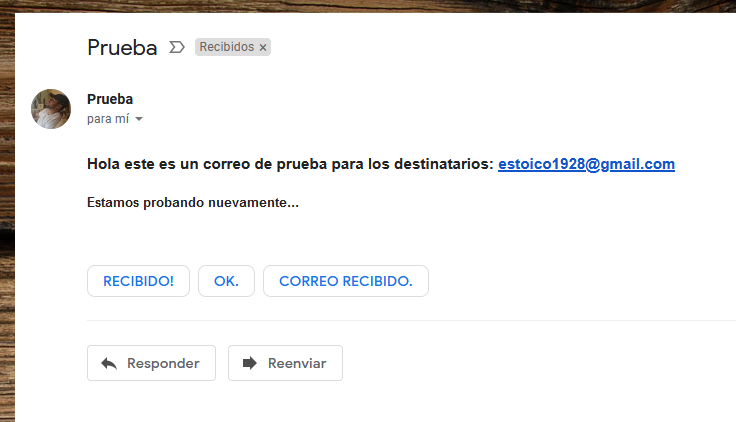


Al SP sp\_NotificacionEmail se le deben pasar dos argumentos, el email y la notificación. Desde el código, el email pasa en una línea a ser parte del *saludo* el cual se almacena en la tabla HistNotificaciones\_Emails junto con el correo, el usuario que ejecuta la acción y la fecha y hora actual. Luego se ejecuta el SP del sistema sp\_send\_dbmail usando algunos de sus parámetros ya pasados a variables.

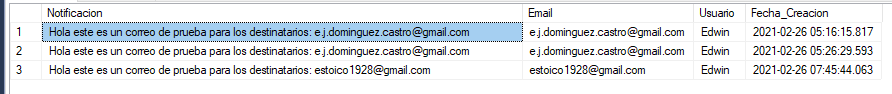
Ésta es su ejecución:



Lo siguiente es el resultado en el correo al que se envió:



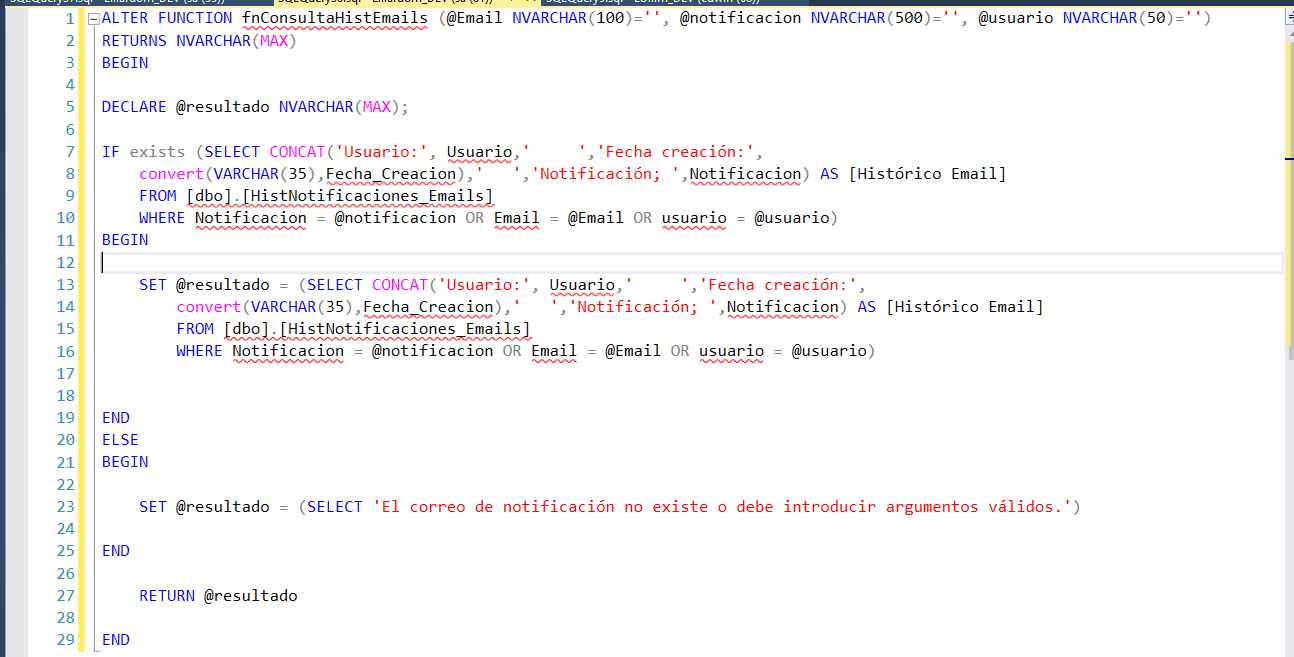
Y lo siguiente es el resultado sobre la tabla HistNotificaciones\_Emails:



El tercer mandato plantea realizar lo siguiente:

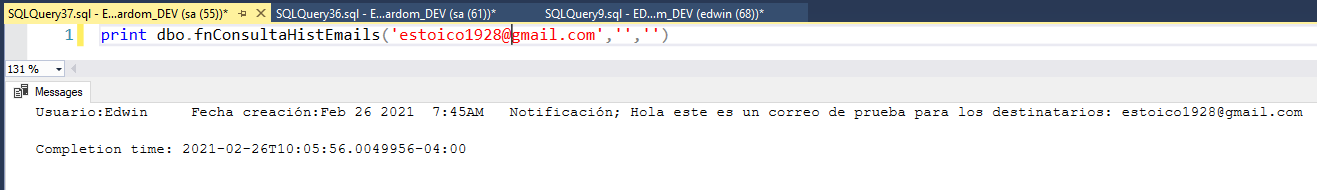
3. Crear función que consulte el historial de envío de emails.

Aquí la función:

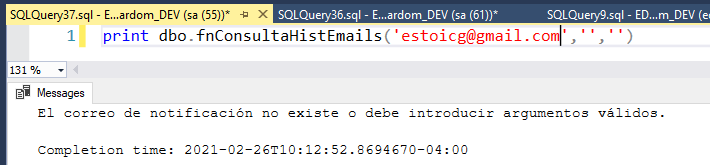


La función fnConsultaHistEmails debe recibir al menos un parámetro de los tres posibles.

Resultado de la función utilizando el parámetro Email:



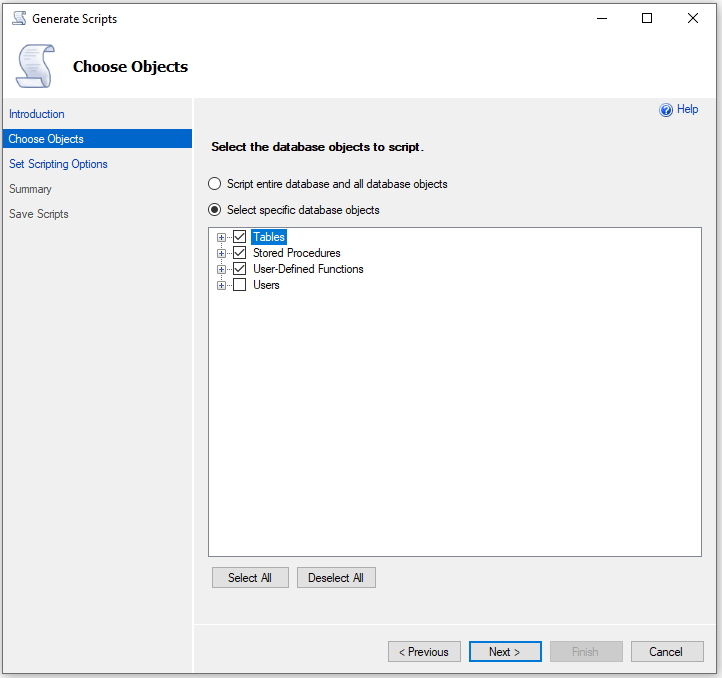
Si el valor en el parámetro es incorrecto, muestra lo siguiente:



El cuarto mandato plantea realizar lo siguiente:

* 1. 4. Se desea que se pueda evidenciar el uso de funciones para la transferencia o cambios de estructura y datos entre una base de datos Dev a una base de datos Productiva. a. Se deben crear dos bases de datos con las siguientes características:
  2. i. Mardom\_DEV
  3. 1. Con las Tablas:
  4. a. Categoría (id, nombre, descripción, fecha creación, fecha actualización, estatus)
  5. b. Productos (id, nombre, estatus, tipo)
  6. c. Tipo Productos (id, tipo, fecha creación, fecha actualización)
  7. d. Compras (idProducto, fecha compra, cantidad, idCategoria, fecha creación, fecha actualización)
  8. ii. Mardom\_PRO
  9. 1. Con las Tablas:
  10. a. Categoría (id, nombre, descripción, fecha creación, fecha actualización, estatus)
  11. b. Productos (id, nombre, estatus, tipo)
  12. c. Tipo Productos (id, tipo, fecha creación, fecha actualización)
  13. d. Compras (idProducto, fecha compra, cantidad, idCategoria, fecha creación, fecha actualización)
  14. iii. Insertar datos de pruebas en las tablas y crear una función para el pase de datos desde una base de datos a otra, así como cambios en el schema de las mismas.

Para esta cuarta situación creé la base de datos de desarrollo con todas los objetos necesarios para un desempeño efectivo (índices,llaves foráneas y otras restrinciones) y solo pude realizar los cambios de estructura y de datos desde la opción del Managment Studio la cual es: sobre la base de datos click derecho->Tasks->Generate Script…



Next…

