

Gestión de Datos

Trabajo Práctico - Migración de Base de Datos

Documento de justificación de decisiones

1c2023

Grupo: ESECUELE

Parte I - Creación de Tablas

Las tablas fueron creadas en base a lo establecido en el Diagrama de Entidad-Relación definido en la entrega previa, sumado a las correcciones correspondientes. Los tipos de datos elegidos se eligieron en correspondencia con aquellos en la tabla original.

El orden de creación de tablas fue elegido de tal manera de que se permitiría definir las Foreign Keys (FK's) en las mismas tablas - por ende el orden fue en base a la cantidad de dependencias que cada tabla tenía. A menor cantidad, más temprano sería creada.

Las tablas en general fueron asignadas claves en base a un entero autogenerado, a fin de facilitar más tarde creación de constraints, triggers y otros objetos - no queríamos estar recordando cuáles eran las PK's naturales de cada objeto y temíamos que resultara demasiado desorganizado si teníamos que agregar, por ejemplo, DNI + NOMBRE + USUARIO para referenciar una cuenta.

Parte 2 - Migración de Datos

El enfoque elegido para la migración fue top-down - empezando con los objetos más genéricos y avanzando hacia aquellos que más referencias tenían a otros objetos. Esto nos permitió poder testear lentamente que todos los objetos previos fueran creados con facilidad antes de aumentar la complejidad.

A continuación, algunas decisiones de implementación importantes.

Generalidades

Para organizar el script, elegimos separarlo en varias partes:

- Los scripts de creación de tablas
- Procedures individuales para la migración tabla-por-tabla
- Ejecución de las procedures

La idea de esto era facilitar la lectura y modificación del script lo más posible.

Creamos una tabla adicional por conveniencia para asignar cada Localidad con su provincia - esta misma es temporal y es eliminada al final del script.

Las constraints fueron definidas en paralelo con la creación de las tablas. Las condiciones de NOT NULL en las foreign key fueron elegidas en base al modelo de negocio - si la agregamos, es que consideramos que ese atributo no podía no estar definido bajo ninguna condición.

Usuario

Para la tabla de *Usuario* se tomaron los valores de la tabla Maestra, identificando a los usuarios por medio de su **DNI, Nombre y Apellido** - esta decisión fue en base de una lectura de los datos disponibles en la tabla existente y por un análisis probabilístico.

Junto a los usuarios, insertamos sus medios de pago y direcciones, al ser valores independientes del resto de las entidades.

Locales

En este caso, el detalle de mayor importancia es que se consideró la combinación de **Nombre y Dirección** como PK para identificar a los *locales* al momento de realizar la migración. Se consideró agregar la localidad como restricción adicional, pero por temas de performance nos quedamos con esos dos valores para no tener que hacer un JOIN para recuperar la localidad.

Las categorías de los locales, a falta de datos en la tabla original, se dejaron todas en NULL - con la opción de ingresar los datos más tarde.

Repartidores y Operadores

Similarmente con el usuario, se tomó el **DNI, Nombre y Apellido** como identificador. Por el momento para el caso del repartidor la localidad *activa* se dejó como un NULL - bajo la posibilidad de que esta pueda ser agregada más tarde empleando otro procedimiento. Una posible opción para la localidad activa es la localidad del último pedido, pero no hay garantía de que esta sea la correcta.

Envío mensajería y Pedido

Nuevamente, al traer los datos utilizamos los identificadores previamente definidos para los usuarios y repartidores. Para las localidades, utilizamos el nombre y la provincia como identificadores. Para los medios de pago, consideramos el número de tarjeta, la marca del tipo de medio de pago (este último quizás no necesario, en base a un análisis de datos de la tabla).

Envíos

Para el caso de los envíos, detectamos la posibilidad de que un pedido tenga más de un pedido - esto podría darse si un envío fuera cancelado por algún tipo de reclamo sobre el pedido, y después fuera reenviado. Por ende, realizamos la corrección correspondiente al modelo, permitiendo una relación 'one to many' en vez de 'one to one' como originalmente habíamos definido

Otras decisiones

Para las tablas de *Pedido*, *Reclamo* y *Cupón*, decidimos reutilizar los números existentes al momento de cargar los datos, y emplear una clave autogenerada para las siguientes entradas a estas tablas. Para lograr esto, desactivamos temporalmente la propiedad que prohíbe la inserción de datos en claves con IDENTITY(1,1).

Debido a la posible existencia de duplicados en la tabla de cupón (entradas con igual número de cupón pero diferentes propiedades), seleccionamos únicamente un solo valor de la tabla - priorizando al usuario más nuevo (mayor ID). Creemos que esto pudo haber sido causado por un error en la base de datos de uno de los integrantes del grupo, y que no hay duplicados en la tabla.

Orden de ejecución

El orden de ejecución fue elegido en base a las interdependencias de las tablas. En general se fue buscando ir desde los agentes (usuarios y sus datos, locales, operadores) y después sus interrelaciones - esto fue simplemente necesario para lograr una correcta ejecución de los procedures y que todos los datos sean tenidos en cuenta.