Práctica Nro. 4
Conceptos aplicados usando MySQL

Fecha de publicación: viernes 26/10/2016
Fecha finalización: viernes 9/11/2016

Introducción

Un taller mecánico con múltiples sucursales posee una base de datos para almacenar información sobre las reparaciones que se realizan para sus clientes, los pagos e informes sobre las mismas.
El esquema con el que cuentan es el siguiente:

REPARACION(codSucursal, dniCliente, fechaInicioReparacion, cantDiasReparacion, telefonoReparacionCliente, direccionReparacionCliente, ciudadReparacionCliente, tarjetaReparacion, nombreApellidoCliente, domicilioCliente, ciudadCliente, tarjetaPrimaria, tarjetaSecundaria, nombreSucursal, domicilioSucursal, ciudadSucursal, encargadoSucursal, m2, empleadoReparacion, repuestoReparacion)

Clave candidata del esquema REPARACION: **Cc**: (dniCliente, fechaInicioReparacion, empleadoReparacion, repuestoReparacion)

Dependencias funcionales válidas en el esquema REPARACION:

df1: dniCliente -> nombreApellidoCliente, domicilioCliente, ciudadCliente, tarjetaPrimaria, tarjetaSecundaria
df2: codSucursal-> nombre, domicilioSucursal, ciudadSucursal, encargadoSucursal, m2
df3: domicilioSucursal, ciudadSucursal -> nombre, codSucursal, encargadoSucursal, m2
df4: dniCliente, fechaInicioReparacion -> codSucursal, cantDiasReparacion, telefonoReparacionCliente, direccionReparacionCliente, ciudadReparacionCliente, tarjetaReparacion
df5: dniCliente, fechaInicioReparacion -> domicilioSucursal, ciudadSucursal, cantDiasReparacion, telefonoReparacionCliente, direccionReparacionCliente, ciudadReparacionCliente, tarjetaReparacion

Luego de haber aplicado el proceso de normalización[1] quedan las siguientes particiones en 4FN:

REPARACION (codSucursal, dniCliente, fechaInicioReparacion, cantDiasReparacion, telefonoReparacionCliente, direccionReparacionCliente, ciudadReparacionCliente, tarjetaReparacion)
CLIENTE (dniCliente, nombreApellidoCliente, domicilioCliente, ciudadCliente, tarjetaPrimaria, tarjetaSecundaria)
SUCURSAL (codSucursal, nombre, domicilioSucursal, ciudadSucursal, encargadoSucursal, m2)
REVISIONREPARACION (dniCliente, fechaInicioReparacion, empleadoReparacion)
REPUESTOREPARACION (dniCliente, fechaInicioReparacion, repuestoReparacion)

Y la siguiente Clave Primaria: (dniCliente, fechaInicioReparacion, empleadoReparacion, repuestoReparacion)

Se proveen dos archivos separados con lo necesario para la creación de las tablas e inserción de datos. Por un lado un esquema normalizado y por otro, uno desnormalizado, llamados **reparacion.sql.zip** y **reparacion_dn.sql.zip** respectivamente.
Para crear los esquemas y cargar los datos, hacerlo desde línea de comando. Para esto, ubicarse con una terminal en el directorio **bin** de la instalación de mysql y ejecutar el siguiente comando para acceder a la terminal mysql:

```
mysql -h localhost -u root -p
```

dentro de la terminal mysql, crear ambos esquemas:

```
mysql> create database reparacion;
mysql> create database reparacion_dn;
mysql> exit;
```

nuevamente en la terminal, cargar los datos en ambos esquemas:

```
mysql reparacion -h localhost -uroot -p < ruta_del_archivo
```

[1] Por brevedad, en este documento se omitió transcribir el proceso de normalización y sólo se muestra las particiones resultantes en 4FN. Para todos los ejercicios planteados en esta materia se debe seguir el proceso de normalización completo.

Donde ***ruta_del_archivo*** es el path al archivo provisto.

Ejercicios

1) Crear usuarios para las bases de datos, usando los nombres reparacion para la versión normalizada y reparacion_dn para la desnormalizada. Habiendo creado estos usuarios, evitar el uso de root para el resto del trabajo práctico.

1.1) Adicionalmente, en ambas bases:
- cree un usuario sólo con permisos para realizar consultas de selección, es decir que no puedan realizar cambios en la base. Use los nombres 'reparacion_select' y 'reparacion_dn_select'.
- cree un usuario que pueda realizar consultas de selección, actualización y eliminación a nivel de filas, pero que no puedan modificar el esquema. Use los nombres 'reparacion_update' y 'reparacion_dn_update'.
- cree un usuario que tenga los permisos de los anteriores, pero que además pueda modificar el esquema de la base de datos. Use los nombres 'reparacion_schema' y 'reparacion_dn_schema'.

2) Listar dni, nombre y apellido de todos los clientes ordenados por dni en forma ascendente. Realice la consulta en ambas bases. ¿Qué diferencia nota en cuanto a performance? ¿Arrojan los mismos resultados? ¿Qué puede concluir en base a las diferencias halladas?

3) Hallar aquellos clientes que para todas sus reparaciones siempre hayan usado su tarjeta de crédito primaria (nunca la tarjeta secundaria). Realice la consulta en ambas bases.

4) Crear una vista llamada 'sucursalesPorCliente' que muestre los dni de los clientes y los códigos de sucursales de la ciudad donde vive el cliente. Cree la vista en ambas bases.

5) En la base normalizada, hallar los clientes que dejaron vehículos a reparar en todas las sucursales de la ciudad en la que viven
   a. Realice la consulta sin utilizar la vista creada en el ej 4.
   b. Realice la consulta utilizando la vista creada en el ej 4.

   Restricción: resolver este ejercicio sin usar la cláusula "NOT EXIST".
   Nota: limite su consulta a los primeros 100 resultados, caso contrario el tiempo que tome puede ser excesivo.

6) Hallar los clientes que en alguna de sus reparaciones hayan dejado como dato de contacto el mismo domicilio y ciudad que figura en su DNI. Realice la consulta en ambas bases.

7) Para aquellas reparaciones que tengan registrados mas de 3 repuestos, listar el DNI del cliente, el código de sucursal, la fecha de reparación y la cantidad de repuestos utilizados. Realice la consulta en ambas bases.

En la base normalizada realice los siguientes ejercicios:

8) Agregar la siguiente tabla:

REPARACIONESPORCLIENTE
idRC: int(11) PK AI
dniCliente: int(11)
cantidadReparaciones: int(11)
fechaultimaactualizacion: datetime
usuario: char(16)

9) Stored procedures
   a) Crear un **stored procedure** que realice los siguientes pasos **dentro de una transacción**:
      - Realizar una consulta que para cada cliente (dniCliente), calcule la cantidad de reparaciones que tiene registradas. Registrar la fecha en la que se realiza la consulta y el usuario con el que la realizó.
      - Guardar el resultado de la consulta en un cursor.
      - Iterar el cursor e insertar los valores correspondientes en la tabla **REPARACIONESPORCLIENTE.**
   b) Ejecute el stored procedure.

10) Crear un **trigger** de modo que al insertar un dato en la tabla **REPARACION**, se actualice la cantidad de reparaciones del cliente, la fecha de actualización y el usuario responsable de la misma (actualiza la tabla **REPARACIONESPORCLIENTE**).

11) Crear un stored procedure que sirva para agregar una reparación, junto con una revisión de un empleado (**REVISIONREPARACION**) y un repuesto (**REPUESTOREPARACION**) relacionados **dentro de una sola transacción**. El stored procedure debe recibir los siguientes parámetros: **dniCliente, codSucursal, fechaReparacion**, **cantDiasReparacion, telefonoReparacion, empleadoReparacion**, **repuestoReparacion**.

12) Ejecutar el stored procedure del punto 11 con los siguientes datos:

dniCliente: **1009443**
codSucursal: **100**
fechaReparacion: **2013-12-14 12:20:31**
empleadoReparacion: '**Maidana'**
repuestoReparacion: '**bomba de combustible'**
cantDiasReparacion: **4**
telefonoReparacion: **4243-4255**

13) Realizar las inserciones provistas en el archivo **inserciones.sql**. Validar mediante una consulta que la tabla **REPARACIONESPORCLIENTE** se esté actualizando correctamente.

14) Considerando la siguiente consulta

```
select count(r.dniCliente) from reparacion r, cliente c, sucursal s, revisionreparacion rv where r.dnicliente=c.dnicliente
and r.codsucursal=s.codsucursal
and r.dnicliente=rv.dnicliente
and r.fechainicioreparacion=rv.fechainicioreparacion
and empleadoreparacion = 'Maidana'
and s.m2<200
and s.ciudadsucursal='La Plata';
```

Analice su plan de ejecución mediante el uso de la sentencia EXPLAIN.

a) ¿Qué atributos del plan de ejecución encuentra relevantes para evaluar la performance de la consulta?
b) Observe en particular el atributo type ¿cómo se están aplicando los JOIN entre las tablas involucradas?
c) Según lo que observó en los puntos anteriores, ¿qué mejoras se pueden realizar para optimizar la consulta?
d) Aplique las mejoras propuestas y vuelva a analizar el plan de ejecución. ¿Qué cambios observa?

15) Análisis de permisos.

a) Para cada punto de la práctica incluido en el cuadro, ejecutarlo con cada uno de los usuarios creados en el punto 1 e indicar con cuáles fue posible realizar la operación.
b) Determine para cada caso, cuál es el conjunto de permisos mínimo.
c) Desde su punto de vista y contemplando lo visto en la materia, explique cuál es la manera óptima de asignar permisos a los usuarios.

Usuario	Permisos asignados	2	3	4	5	6	7	8	9a	9b	10	11	12	13	14	15a	15b
reparacion																	
reparacion_dn																	
reparacion_select																	
reparacion_dn_select																	
reparacion_update																	
reparacion_dn_update																	
reparacion_schema																	
reparacion_dn_schema																	