



Práctica Nro. 4 Conceptos aplicados usando MySQL

Fecha de publicación: jueves 2/10/2014

Fecha límite de entrega -carácter obligatorio-: 24/10/2014

La aprobación de este trabajo en primera fecha, junto a la aprobación de todos los temas del parcial en primera fecha son condición necesaria para acceder a la promoción de la materia.

En caso de desaprobar este trabajo en la primera entrega, habrá una fecha de re entrega durante diciembre de 2014.

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 normalización¹ quedan los siguientes esquemas 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)

¹ Por brevedad, en este documento se omitió transcribir el proceso de normalización y sólo se muestra el resultado. En los ejercicios a realizar la cursada no se debe omitir.



REPUESTOREPARACION (dniCliente, fechaInicioReparacion, repuestoReparacion)

Se proveen dos archivos separados con lo necesario para creación de tablas e inserción de datos. Por un lado un esquema normalizado y por otro uno desnormalizado.

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:

```
mysql reparacion -h localhost -u root -p pass_root < ruta_del_archivo
```

Donde *pass_root* es la contraseña del usuario root y *ruta_del_archivo* es el path más el nombre del archivo provisto.

Ejercicios

- 1) Crear usuarios para las bases de datos. Estos usuarios son los que se deben usar para realizar el resto del trabajo práctico.

Para hacer esto en la base de datos normalizada:

```
GRANT USAGE ON *.* TO 'reparacion'@'%' IDENTIFIED BY 'reparacion';  
GRANT ALL PRIVILEGES ON `reparacion`.* TO 'reparacion'@'%' WITH GRANT OPTION;
```

En la base de datos desnormalizada:

```
GRANT USAGE ON *.* TO 'reparacion_dn'@'%' IDENTIFIED BY 'reparacion';  
GRANT ALL PRIVILEGES ON `reparacion_dn`.* TO 'reparacion_dn'@'%' WITH GRANT OPTION;
```

- 2) Listar los dni, nombre y apellido de todos los clientes ordenados por dni en forma ascendente. Realice la consulta en ambas bases. ¿Qué diferencia nota?
- 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. Realice 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.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) 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**.

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**.

14) Validar mediante una consulta que la tabla **REPARACIONESPORCLIENTE** se este actualizando correctamente

15) A continuación se muestran las sentencias necesarias para visualizar el plan de ejecución de ciertas consultas sobre la base de datos normalizada - **reparacion**. El objetivo es que luego de correr los scripts dados, usted analice los resultados y escriba las conclusiones a las que llega.

Ejecutar en el orden dado y para cada caso guardar en un archivo el resultado.

Plan 1)

```
explain extended
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';
```

Plan 2)



```
explain extended
select r.*
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 = 'Sosa'
and s.m2>80
and s.ciudadsucursal='Vicente Lopez'
and r.fechainicioreparacion between '2006-01-01' and '2013-12-31'
;
```

Plan 3)

```
explain extended
select r.*
from reparacion r, cliente c, revisionreparacion rv,
(select codsucursal from sucursal where ciudadsucursal='Vicente Lopez' and m2>100)
s
where r.dnicliente=c.dnicliente
and r.codsucursal=s.codsucursal
and r.dnicliente=rv.dnicliente
and r.fechainicioreparacion=rv.fechainicioreparacion
and empleadoreparacion = 'Dominguez'
and r.fechainicioreparacion between '2006-01-01' and '2013-12-31';
```

Luego, crear los siguientes índices:

```
create index idx_m2 on sucursal(m2);
create index idx_er on revisionreparacion(empleadoreparacion);
create index idx_ciu_suc on sucursal(ciudadsucursal);
```

Una vez creados los índices ejecutar nuevamente los planes 1-2-3 y guardar los resultados a archivo

Para cada plan, determinar cual es la consulta involucrada

Realizar un análisis comparativo de cada plan con y sin índices, indicando donde cree que incide la incorporación del índice en relación a la variación en el plan de ejecución.

Para cada análisis elaborar una conclusión.

- 16) Las siguientes sentencias muestran los planes de ejecución para las consultas equivalentes del ejercicio 13 pero, en este caso, sobre la base de datos sin normalizar – **reparacion_dn**. Nuevamente, el objetivo es que luego de correr los scripts dados, usted analice los resultados y escriba las conclusiones a las que llega.

Plan A)

```
explain extended
select count(*) from reparacion r
where empleadoreparacion = 'Maidana'
and r.m2<200
and r.ciudadsucursal='La Plata'
;
```

Plan B)

```
explain extended
select * from reparacion r
where r.empleadoreparacion = 'Sosa'
and r.m2>80
and r.ciudadsucursal='Vicente Lopez'
and r.fechainicioreparacion between '2006-01-01' and '2013-12-31'
```



;

Plan C)

```
explain extended
select * from reparacion r
where empleadoreparacion = 'Dominguez'
and r.m2>100
and r.ciudadsucursal='Vicente Lopez'
and r.fechainicioreparacion between '2006-01-01' and '2013-12-31'
;
```

Luego, crear los siguientes índices:

```
create index idx_canthab on reparacion(m2);
create index idx_dr on reparacion(empleadoreparacion);
create index idx_ciu_hosp on reparacion(ciudadsucursal);
```

Una vez creados los índices ejecutar nuevamente los planes A-B-C y guardar los resultados a archivo

Para cada plan, determinar cual es la consulta involucrada

Realizar un análisis comparativo de cada plan con y sin índices, indicando dónde cree que incide la incorporación del índice en relación a la variación en el plan de ejecución.

Para cada análisis elaborar una conclusión.