Universidad del Valle

Facultad de Ingeniería
Escuela de Ingeniería de Sistemas y Computación

Bases de Datos I - Laboratorio #3 – SQL 2: Subconsultas, Métodos Para Mantener la Integridad Referencial – Manejo de Restricciones, Vistas

Entrega: Noviembre 23 de 2017

NORMAS PARA LA ENTREGA DE LOS LABORATORIOS

a. *Coloque su nombre y dirección de correo electrónico preferida en la carátula de su informe, así como el nombre del profesor a quien le entrega el informe. Los informes deben estar firmados por todos los integrantes del grupo, debajo del siguiente código de ética:*

b. ***<< Al firmar el presente informe, aseguramos que nuestro grupo NO ha copiado de nadie, ni dado copia a nadie, la solución que a continuación presentamos>>***

c. *Coloque el nombre de los integrantes del grupo, nombre del profesor, número del grupo de Bases de Datos y número de laboratorio presentado. Sin esta información en la carátula no se recibirá su informe.*

d. *Organice su trabajo en carpetas, en lo posible una por cada punto del laboratorio.*

e. *Imprima una copia del código fuente de todos sus programas y entréguelo en el informe. Recuerde que es responsabilidad total y absoluta de los integrantes del grupo el verificar la existencia de la información correspondiente al informe en el medio de almacenamiento. No se aceptan reclamos por un error en la grabación de los datos del laboratorio.*

f. *Evidencia de la ejecución. Esto se puede realizar copiando y pegando, en el informe, las ventanas donde se suministra información o donde se entrega información por el programa.*

g. ***Lugar, Plazo y Medio de Entrega**: Todos los archivos que se soliciten en el informe deben ser colocados en un archivo comprimido (.zip) del laboratorio que se está resolviendo. Presente su informe en la fecha indicada, descargándolo en el Campus Virtual.*

h. *Durante el curso no se recibirán informes de laboratorio enviados por correo electrónico,*

Objetivos:

i. Actualizar datos en una base de datos
ii. Describir los métodos para mantener la integridad referencial en las Bases de Datos
iii. Implementar métodos alternativos de integridad referencial proporcionados por el DBMS
iv. Identificar las ventajas de implementar las restricciones en los datos a nivel del DBMS
v. Aplicar las restricciones necesarias a un esquema de BD.
vi. Utilizar vistas para el manejo de datos.

Metodología:

Se debe elaborar un informe del trabajo realizado, para ello deberá describir cada una de las actividades que se realizó y responder a las preguntas formuladas. El laboratorio debe ser desarrollado en grupos de máximo 3 personas.

Se debe presentar un informe con las instrucciones SQL para cada punto y además una o varias imagen(es) como evidencia de la ejecución, por cada punto, que incluyan el borde de la ventana.

1. Aspectos Básicos I

Actualización de Información

Para la actualización de los datos existentes en la base de datos, se pueden usar las sentencias estándar para borrado (DELETE) y actualización (UPDATE) de datos.

La primera permite la eliminación de una o más filas de una tabla, y su sintaxis es la siguiente:

```
DELETE FROM [nombre de la tabla]
WHERE [condiciones para el borrado];
```

Digite:

```
DELETE
FROM Titulo
WHERE codigoTit = 'L15';
```

El resultado de la instrucción anterior será la eliminación del registro que corresponde al Título del libro con código L15. Puede verificar lo anterior?

Para realizar la actualización de los datos de un registro, se utiliza la sentencia **UPDATE** la cual permite modificar los valores de las columnas en una o más filas de una tabla, su sintaxis de uso es:

```
UPDATE [nombre de la tabla]
SET [atributo a cambiar = nuevo valor, atributo a cambiar = nuevo valor,...]
WHERE [condición para ejecutar la actualización];
```

Digite la instrucción:

```
UPDATE Titulo
SET autorTit = 'Otro Autor'
WHERE codigoTit = 'L14';
```

Ahora realice una consulta sobre la tabla modificada para que observe que sucede al ejecutarse la instrucción anterior, así:

```
SELECT * FROM Titulo
WHERE codigoTit = 'L14';
```

Se podrá observar que ahora el nuevo precio de la asignatura con autor del libre es `'Otro Autor'`.

Trabajo #1 (12 puntos)

Escriba las sentencias SQL para hacer las siguientes operaciones:

1. [3] Modificar el nombre del editor, para la editorial con código = E3 a 'Nueva Editorial'
2. [3] Modificar el país del autor o con nombreAutor = 'AT4' al valor 'Mexico'
3. [3] Eliminar las copias del libro con código L10
4. [3] Borrar el estudiante con código 200622555

Recuerde que debe crear los datos que permitan dar una respuesta a las consultas. Consulta sin respuesta no se tendrá en cuenta.

1. Aspectos Básicos I

Funciones Agregadas

Función	significado	Sintaxis de uso
SUM	Suma valores de una columna	`SELECT atributo(s), función(columna) FROM tabla(s) WHERE (condición) GROUP BY columna;`
COUNT	Cuenta registros	
MAX	Halla el máximo valor de una columna	
MIN	Halla el valor mínimo de una columna	
AVG	Halla el valor promedio de una columna	

Para la realización de funciones un requerimiento muy común es la agregación de datos, esto se realiza a través de la cláusula GROUP BY. Esta cláusula agrupa las filas con iguales valores en las columnas indicadas después de esta, formando así subconjuntos. Así, todos los atributos dentro del SELECT deben estar en el GROUP BY, un ejemplo de esto sería:

Obtener el número de copias adquiridas, por cada libro:

```
SELECT codigoTit, SUM(nroCopia)
FROM Copia
GROUP BY codigoTit;
```

De esta forma primero se agrupan las copias por cada libro, obteniendo subconjuntos de quienes tienen el mismo `codigoTit`, y luego la función **`SUM`**, que suma el número de copias, para cada uno de los grupos formados.

Trabajo #2 (5 puntos)

5. [5] Realice una consulta con cada una de las funciones agregadas del SQL.

Subconsultas en SQL

Las sentencias **`SELECT`** pueden "anidarse" unas dentro de otras, usando en las más externas el resultado de las más internas. Una subconsulta provee como resultado un conjunto de valores, que puede ser usado como argumento de búsqueda en un predicado de pertenencia IN o EXISTS. Veamos los siguientes ejemplos básicos de una subconsulta:

Predicado IN

```
SELECT nombreEst, planEstudio, telefonoEst
FROM Estudiante
WHERE codigoEst IN
    (SELECT codigoEst
     FROM Prestamo
     WHERE codigoTit IN
         (SELECT codigoTit
          FROM Titulo
          WHERE codEditorial = “E1” ) );
```

Consulta que muestra el nombre y apellido de todos los empleados que tienen asignada una actividad dentro de cualquier proyecto

Predicado EXISTS

```
SELECT nombreEst, planEstudio, telefonoEst
FROM Estudiante
WHERE EXISTS
    (SELECT *
     FROM Prestamo
     WHERE Prestamo.codigoEst =
           Estudiante. codigoEst );
```

Consulta que muestra el nombre, plan de estudio y número telefónico de los estudiantes que han realizado préstamos de libros.

Algunas de las consultas que se plantean a continuación exigen que se prepare un “paquete de datos” que permita comprobar el funcionamiento de la consulta. Asegúrese de ingresar los datos para poder demostrar que la consulta funciona correctamente.

Trabajo #3 (6 puntos)

En el entorno SQL desarrolle las siguientes consultas, utilizando los predicados IN o EXISTS:

6. Ahora tomando como base el ejercicio del laboratorio 2, por medio de una subconsulta, obtenga los libros editados por algún editor de la ciudad de Cali. [2]
7. Usando subconsultas, obtenga los libros que han sido adquiridos a costo mayor al del promedio del costo de todos los libros. [2]
8. Usando subconsultas, obtenga la información de los estudiantes de ingeniería civil que han prestado todos los libros del autor AT1. [2]

Recuerde que debe crear los datos que permitan dar una respuesta a las consultas. Consulta sin respuesta no se tendrá en cuenta.

2. Aspectos Básicos II

Vistas

Una vista es un objeto o una tabla lógica que presenta datos de una o más tablas; se puede usar para mostrar todas las filas y columnas de una tabla, de modo que para motivos de seguridad proteja el nombre de su tabla de origen, también puede mostrar un subconjunto de filas y columnas de una o más tablas, permitiendo proteger algunos datos de ciertas tablas. Otra de sus formas de uso es para simplificar la codificación de aplicaciones, pues con estas puede unir datos de diferentes tablas y hacerlas ver como si estuvieran relacionadas, o mostrar datos derivados que no están realmente en una tabla especifica.

La sintaxis básica para crear una vista es:

```
CREATE VIEW nombre_vista AS
SELECT instrucciones...
FROM nombre_Tabla | nombre_vista
[WHERE (condición)];
```

Digite la instrucción:

```
CREATE VIEW vistaEditorial AS
SELECT codEditorial , nombreEditor, sedeEditor
FROM EDITOR;
```

Ahora realice una consulta sobre la vista que se acaba de crear. Muestre en su informe el resultado.

Trabajo #4 (6 puntos)

Digite la instrucción:

```
CREATE VIEW vistaEditorial AS
SELECT codEditorial , nombreEditor, sedeEditor
FROM EDITOR;
```

9. Ahora realice una consulta sobre la vista que se acaba de crear. Muestre en su informe el resultado. [2]
10. Cree una vista (llámela DatosTitulos) de las copias de títulos, con el código del título u obra, número de la copia, nombre del Titulo, el autor y el nombre de la editorial. [2]
11. Cree una vista de los estudiantes que contenga los datos del estudiante sin la dirección ni teléfono de cada estudiante. Asígnele el nombre VistaEstudiante como nombre a la vista. [2]

Recuerde que debe crear los datos que permitan dar una respuesta a las consultas. Consulta sin respuesta no se tendrá en cuenta.

Aspectos Básicos III

Restricciones

En la realización de un esquema se pueden necesitar algunas restricciones que permitan o den la posibilidad de tener un control sobre los datos que se ingresaran en la base de datos, de modo que no atenten contra la integridad ya establecida de los datos.

NOT NULL: La columna no permitirá valores nulos.

CONSTRAINT: Permite asociar un nombre a una restricción.
[CONSTRAINT <nombreRestriccion>] CHECK (condicion)

DEFAULT valor: La columna tendrá un valor por defecto. El DBMS utiliza este valor cuando no se especifica un valor para dicha columna. Se puede especificar dentro de un constraint.

PRIMARY KEY (columna1 [, columna2...]): Permite indicar que una o varias columnas forman parte de la clave primaria.

UNIQUE: Obliga a que los valores de una columna (o varias) tomen valores únicos (no puede haber dos filas con igual valor en dichas columnas).

FOREIGN KEY: Permite declarar una llave foránea o clave externa y hace referencia a la clave primaria de otra tabla. Otra forma es declarar la palabra REFERENCES después del atributo. Sintaxis:
FOREIGN KEY (A1, , ... , Aj) REFERENCES R (K1, ... , Kj)
[ON DELETE {SET DEFAULT | SET NULL | CASCADE | NO ACTION}]
[ON UPDATE {SET DEFAULT | SET NULL | CASCADE | NO ACTION}]

CHECK (condición): Permite indicar que condición deben de cumplir uno o varios campos de la tabla.

DOMINIO: Permite definir un nuevo tipo de dato, especificando la restricción sobre los valores posibles que puede tomar un campo de este tipo: **CREATE DOMAIN** nombreDominio tipoDato **CHECK** (valores permitidos);

Problema del Curso: Alquiler de coches "Autos UV" (45 puntos)

"Autos UV" necesita diseñar una base de datos para almacenar y gestionar la información empleada por la empresa que se dedicada al alquiler de automóviles; por ello deben tenerse en cuenta los siguientes aspectos:

- La empresa dispone de un conjunto de coches para su alquiler. Se necesita conocer la placa, marca y modelo, el color y el precio de alquiler de cada coche. Además cada coche está asignado a un determinado garaje, que no puede cambiar. De cada garaje interesa conocer su código, denominación o nombre y dirección. La empresa dispone de varias sedes (agencias) ubicadas en diferentes sitios.
- Los datos que interesa recoger de cada cliente son el NIT, nombre, dirección, ciudad y número de teléfono; además, los clientes se diferencian por un código interno de la empresa.
- Si un cliente desea solicitar el alquiler de algún vehículo, se pone en contacto con una de las agencias de la empresa de alquiler "Autos UV" y realiza una reserva. De cada agencia interesa conocer su código y nombre.
- Un mismo cliente puede haber realizado varias reservas a lo largo del tiempo, y por supuesto algunas de esas reservas pueden tener sus fechas traslapadas. Como hemos indicado, en cada reserva participa una agencia, que no tiene que ser la misma para las distintas reservas de un mismo cliente.
- Una reserva puede incluir uno o varios coches y cada reserva tiene un número (código) asignado. De cada reserva interesa registrar la fecha de inicio (cuándo se deben entregar los vehículos al cliente), la fecha de finalización (la fecha prevista de devolución por parte del cliente), el precio total de la reserva y un indicador de si los coches han sido devueltos. Además, para cada vehículo interesa conocer los galones de combustible que contiene en el momento de su entrega al cliente. El precio final de reserva de cada coche se obtiene multiplicando su precio de alquiler por los días que el cliente desea reservarlo. El precio total de la reserva se obtiene sumando los precios finales de alquiler de los coches que incluye dicha reserva.

En este problema, usted llevará a cabo la implementación de la base de datos relacional para "AutosUV" basado en el esquema dado. Para ello se sugieren los siguientes pasos:

i. Prepare su instrucción (SQL) usando el **editor de texto** de pgAdmin.
ii. Si la instrucción **trabaja** sálvela. De otra forma, ubique el error, corríjala e intente de nuevo.
iii. **Salve** su archivo de texto periódicamente antes de perder lo que ha digitado.

Esquema Relacional de la base de datos "Autos UV":

Garaje (codigo, nombre, dirección)

Vehiculo (placa, marca, modelo, color, garaje, precioAlquiler)

LLAVE FORANEA garaje REFERENCIA codigo EN Garaje

Cliente (nitCli, nombre, dirección, ciudad, teléfono)

Agencia (codigo, nombre, dirección, ciudad)

Reserva (numeroR, fechaIni, fechaFin, cliente, agencia, precioR, FecDevolucion)

LLAVE FORANEA agencia *REFERENCIA* codigo *EN Agencia*

LLAVE FORANEA cliente REFERENCIA nitCli *EN Cliente*

ListaReserva (reserva, placaVehic, litrosInicio, precioReserva)

LLAVE FORANEA reserva REFERENCIA reserva EN Reserva

LLAVE FORANEA placaVehic REFERENCIA placa EN Vehiculo

Usted(es) debe(n) de crear la base de datos, poblarla con datos significativos (que permitan dar respuesta a las consultas abajo planteadas). Por ello se requiere:

- Presentar el esquema SQL
- Presentar los datos ingresados en cada una de las tablas
- Presentar cada una de las consultas planteadas usando SQL
- Mostar una evidencia (imagen con pantallazo) de la ejecución. Sin este requisito no se califica el punto, aunque presente la consulta o instrucción. Imagen de datos sin la instrucción correspondiente no se califica.

Trabajo #5: Manejo de Restricciones *(29 pts)*

Asegúrese de implementar todas las restricciones de integridad necesarios para la base de datos:

12. [2] Asigne el valor por defecto $185.000 al atributo `precioReserva` en la tabla **`ListaReserva`**.
13. [3] Cree un dominio para el sexo con los posibles valores 'Masculino' o 'Femenino'. Asigne al campo sexo del Cliente este tipo de dato. Se puede hacer en POSTGRES? Documente su respuesta.
14. [2] El atributo `marca` de la tabla `Vehiculo` solo puede tomar los siguientes valores: `Chevrolet, Mazda, Renault, Mitsubishi, Volkswagen, Ford, Volvo, GMC, Dodge, Hunday, Toyota, Nissan, BMW, Fiat`. El valor por defecto es "`Desconocido`".
15. [2] Asegúrese que en la tabla `Cliente` la edad esté en el rango 18 y 55 años.
16. [2] La fecha de inicio del alquiler del vehículo debe ser por defecto la fecha actual (de hoy).
17. [4] En la tabla `Reserva` especificar que si un cliente es retirado de la tabla `Cliente` todos los registros de `Reserva` serán eliminados, y si se modifica el cliente, todos los registros de `Reserva` deben ser actualizados al nuevo valor.
18. [3] En la tabla `Vehiculo` especificar que si un garaje es retirado de la tabla `Garaje` todos los registros de `Vehiculo` se les asigne valor nulo, y si se modifica el garaje, todos los registros de `Vehiculo` deben ser actualizados a valor nulo.
19. [4] Crear una restricción que permita verificar que el usuario no podrá introducir una fecha de fin de la reserva del vehículo (`fechaFin`) menor que la fecha de inicio del alquiler (`fechaIni`).
20. [4] Cargue en todas las tablas de "Autos UV", con al menos 20 registros, necesarios para responder las consultas planteadas, usando el comando INSERT. Asegúrese que sus inserciones no violen el "sentido común" ni las restricciones planteadas. Liste los datos ingresados.
21. [2] Cambie el tipo de dato de la columna `precioReserva` en la tabla **`ListaReserva`** a Double(12,2) usando el comando ALTER.
22. [1] Adicione la restricción de dominio >5 y <10 para el atributo `galonesInicio` en la tabla **`ListaReserva`** usando el comando ALTER.

Trabajo #6: Creación de la Base de Datos *(6 pts)*

23. [6] Crear la base de datos, poblarla con datos significativos (que permitan dar respuesta a las consultas abajo planteadas). Por ello se requiere:
    - Presentar el esquema SQL
    - Presentar los datos ingresados en cada una de las tablas

Trabajo #7: Consultas Sencillas (15 pts)

- Presentar cada una de las consultas planteadas usando SQL
- Mostar una evidencia (imagen con pantallazo) de la ejecución. Sin este requisito no se califica el punto, aunque presente la consulta.

Se recomienda resolver estas consultas, cuya dificultad es baja y media, antes de intentar resolver las consultas siguientes:

24. [3] Listar las reservas realizadas por los clientes de la ciudad de 'Jamundí'.
25. [3] Resumen de las reservas (numero, fechaIni, fechaFin, precioTotal) pendientes de devolución, ordenado por fecha de finalización.
26. [3] Listar las agencias (código, nombre) que participan en reservas aún no devueltas.
27. [3] Agencias (código, nombre, ciudad) que no participan en ninguna reserva desde el 15 de enero de 2015, a la fecha.
28. [3] Garajes (código, nombre, dirección) que tienen vehículos que se han reservado a clientes de la ciudad de Palmira.

Trabajo #8: Otras Consultas (21 pts)

29. [3] Valor total de las reservas en las que ha participado cada vehículo en el primer semestre de 2016.
30. [3] Clientes (nit, nombre) con más de 3 reservas, alguna de las cuales se haya realizado mediante la agencia con código A2.
31. [3] Vehículos (matrícula, marca, modelo) reservados más de 3 veces.
32. [3] Vehículos (matrícula, marca, modelo) que más veces ha sido reservado, indicando cuántas veces.
33. [3] Clientes (nit, nombre) que hayan reservado algún coche de la marca 'Volkswagen' más de 2 veces.
34. [3] Suma total del precio de las reservas realizadas por clientes que hayan hecho menos de 3 reservas.
35. [3] Clientes (nit, nombre) que han hecho reservas de todas las agencias.

Usted necesita presentar en su informe:

La copia impresa de los comandos en **POSTGRES** y los resultados (las respuestas de **POSTGRES**) de la ejecución de tales comandos. No incluya comandos con errores, con ello NO obtendrá créditos en su nota.

Antes de finalizar este ejercicio, asegúrese de salvar todos los comandos en un archivo (un archivo script llamado **`autosUV.sql`**). Este archivo servirá como el esquema definitivo del problema.

Recuerde que debe crear los datos que permitan dar una respuesta a las consultas. Consulta sin respuesta no se tendrá en cuenta.

Fecha de Entrega: **Noviembre 23 de 2017**