**LABORATORIO 3**

**GESTIÓN DE BASES DE DATOS**

**KEVIN STEVEN GARCIA – 1533173, kevin.chica@correounivalle.edu.co**

**DIANA CAROLINA ARIAS - 1528008, arias.diana@correounivalle.edu.co**

**SERGIO ALEJANDRO JARAMILLO – 1531614, jaramillo.sergio@correounivalle.edu.co**

**CESAR ANDRES SAAVEDRA - 201628466**

**MAURICIO FERNANDEZ**

**UNIVERSIDAD DEL VALLE**

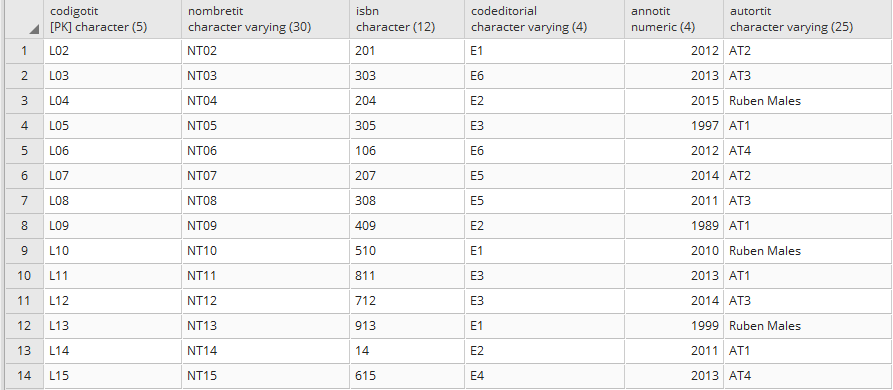
**NOVIEMBRE 2017**

**Trabajo #1 (12 puntos)**

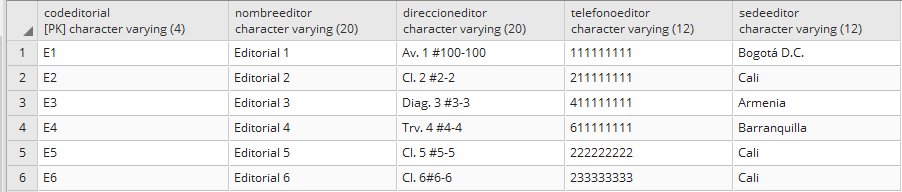
**Escriba las sentencias SQL para hacer las siguientes operaciones:**

Antes de hacer las respectivas modificaciones, mostraremos nuestras tablas originales:

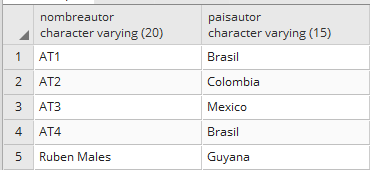
* **Tabla Titulo**:



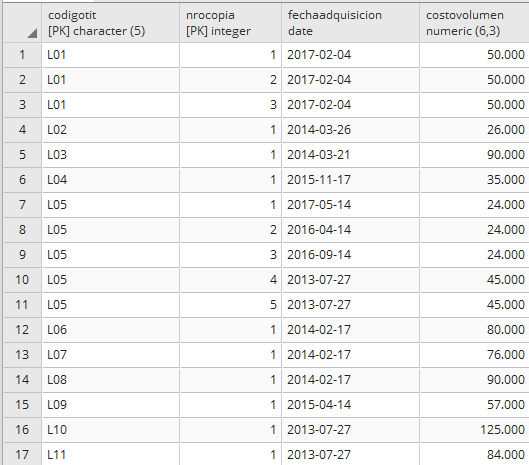
* **Tabla Editor:**

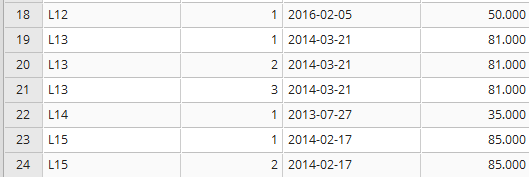
****

* **Tabla Autor:**

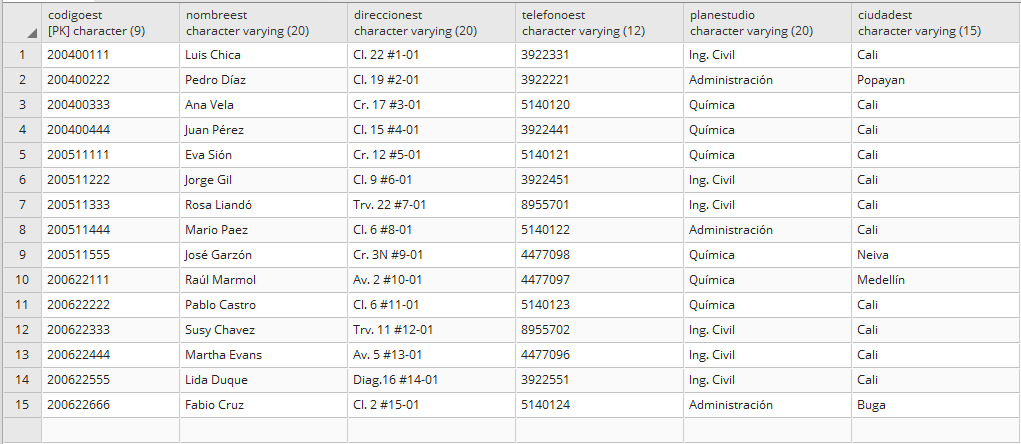
****

* **Tabla Copia:**

****

****

* **Tabla Estudiante:**

****

* **Tabla Prestamos:**

****

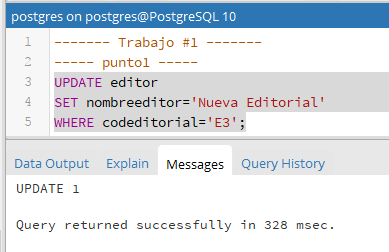
**1. [3] Modificar el nombre del editor, para la editorial con código = E3 a ‘Nueva Editorial’**

Para el desarrollo de este punto se utilizó el código:

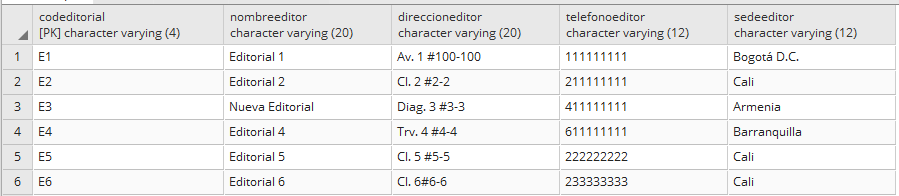
UPDATE editor

SET (nombreeditor='Nueva Editorial')

WHERE (codeditorial='E3');



En esta imagen podemos ver el código en funcionamiento, además vemos el mensaje de confirmación que nos dice que fue actualizada una tupla.

****

En esta imagen podemos ver que se hizo satisfactoriamente la actualización, y el nombre del editor, para la editorial con código E3, paso de ser ‘Editorial 3’ a ‘Nueva Editorial’.

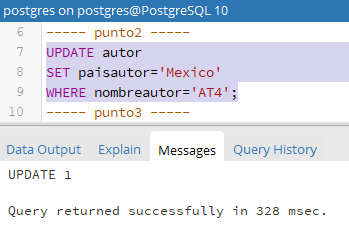
**2. [3] Modificar el país del autor o con nombreAutor = ‘AT4’ al valor ‘Mexico’**

Para hacer la modificación correspondiente, se utilizó el código:

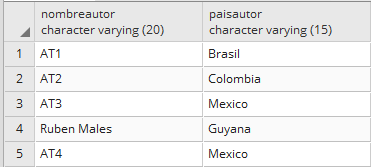
UPDATE autor

SET paisautor='Mexico'

WHERE nombreautor='AT4';

****

En esta imagen podemos ver el código utilizado, y observamos el mensaje de confirmación que nos dice que se modificó o se actualizó una tupla.

****

En esta otra imagen, vemos que el código funcionó correctamente, modificándonos el país del autor con nombre ‘AT4’ de ‘Brasil’ a ‘Mexico’.

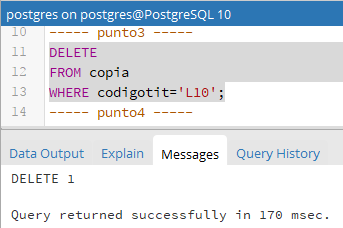
**3. [3] Eliminar las copias del libro con código L10**

Este punto se realizó con el siguiente código:

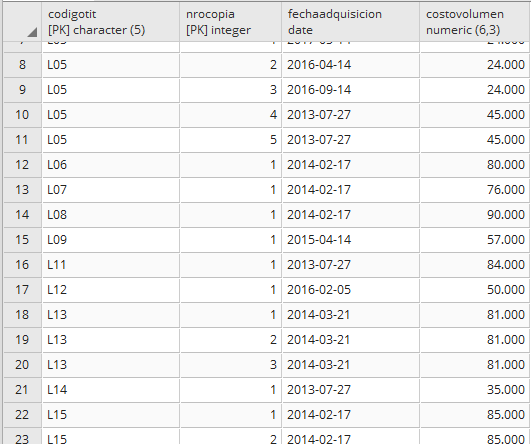
DELETE

FROM copia

WHERE codigotit='L10';

****

En la anterior imagen podemos ver el código utilizado y el mensaje de confirmación que nos indica que una tupla ha sido eliminada.



En esta imagen podemos ver que obtenemos lo que se busca con el código. Se eliminó las copias con código L10, en este caso solo había una.

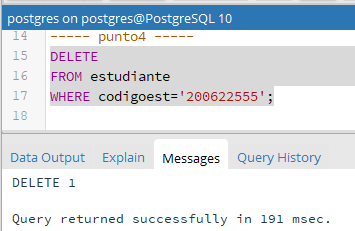
**4. [3] Borrar el estudiante con código 200622555**

Para borrar a dicho estudiante, se utilizó el código:

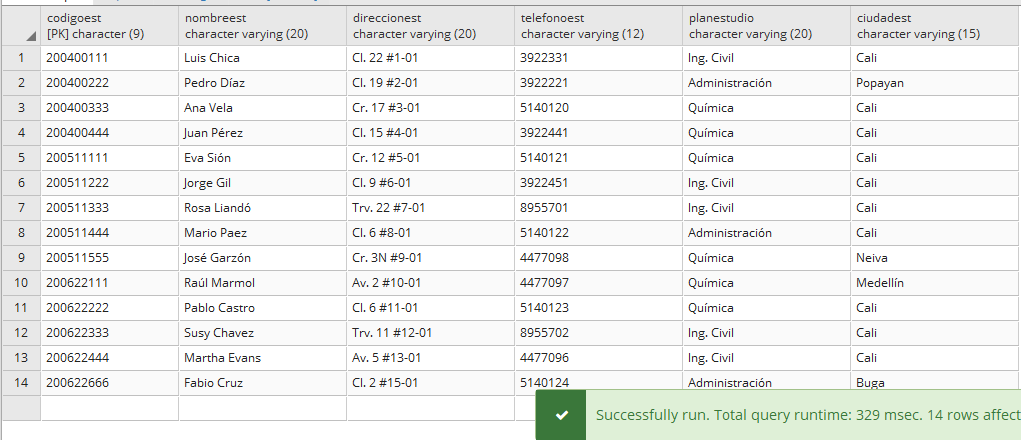
DELETE

FROM estudiante

WHERE codigoest='200622555';



En la imagen anterior podemos ver nuestro código en funcionamiento, observamos el mensaje de confirmación que nos dice que ha sido eliminada una tupla.



En esta imagen podemos ver que la tupla donde el atributo codigoest es 200622555 ha sido eliminada de la tabla Estudiante.

**Trabajo #2 (5 puntos)**

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

* FUNCIÓN SUM:

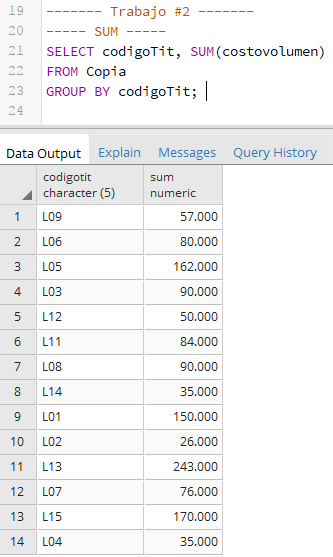
Para realizar una consulta con esta función, decidimos preguntarnos por el costo total por copia o título, es decir, queremos saber cuánto dinero hay, por cada código de obra.

Para responder a dicha cuestión utilizamos el siguiente código:

SELECT codigoTit, SUM(costovolumen)

FROM Copia

GROUP BY codigoTit;



En la anterior imagen podemos ver que el código utilizado funciona perfectamente, y obtenemos el costo o el valor total de copias por cada título.

* Función COUNT:

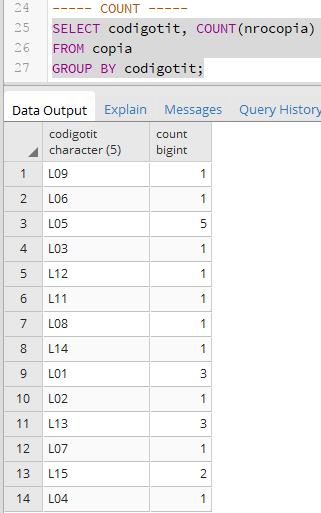
Para elaborar una consulta con esta función, decidimos dar respuesta a la pregunta, ¿Cuántas copias hay por título?

Para dar con este resultado, utilizamos el siguiente código:

SELECT codigotit, COUNT(nrocopia)

FROM copia

GROUP BY codigotit;



En esta imagen podemos ver que el resultado del código utilizado es el deseado, y obtenemos el número de copias que hay de cada libro o título.

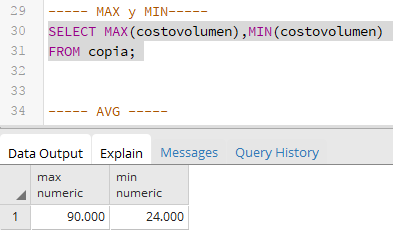
* Función MAX y MIN:

Para utilizar estas dos función, quisimos responder a las siguientes dos preguntas, ¿Cuál es el costo máximo de una copia de un título? y, ¿cuál es el costo mínimo?

Para dar respuestas a estas dos preguntas se utilizó el siguiente código:

SELECT MAX(costovolumen),MIN(costovolumen)

FROM copia;



En esta imagen podemos ver que el código funciona perfectamente, arrojándonos que la copia más cara tiene un valor de 90.000 y la más barata tiene un valor de 24.000. Note que se utilizaron las funciones de agregación MAX y MIN en un mismo código.

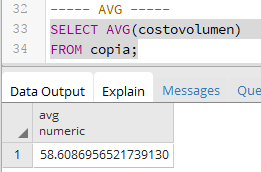
* Función AVG:

Para hacer una consulta con esta función, nos hicimos la pregunta, ¿Cuál es el costo promedio de cada copia?

Para dar respuesta a ello, dimos uso al siguiente código:

SELECT AVG(costovolumen)

FROM copia;



En esta imagen logramos ver el perfecto funcionamiento de nuestro código, arrojándonos el costo o el valor promedio de cada copia, que es de aproximadamente 58608,6956.

**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.**

Para obtener los libros editados por algún editor de la ciudad de Cali por medio de una subconsulta, utilizamos el siguiente código:

SELECT codigotit

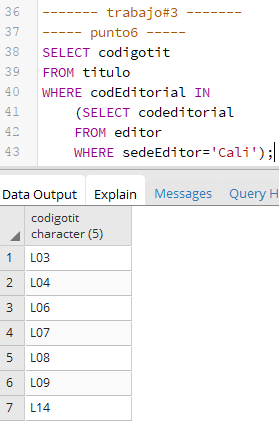
FROM titulo

WHERE codEditorial IN

(SELECT codeditorial

FROM editor

WHERE sedeEditor='Cali');



En esta imagen podemos ver nuestro código en funcionamiento, y observamos que funciona perfectamente, ya que nos arroja los códigos de los títulos que son editados por un editor con sede en la ciudad de Cali.

**7. Usando subconsultas, obtenga los libros que han sido adquiridos a costo mayor al del promedio del costo de todos los libros.**

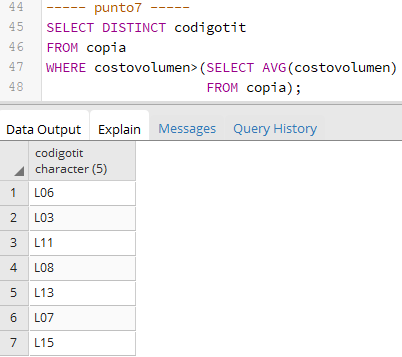
Para dar solución a esta consulta por medio de subconsultas, utilizamos el código siguiente:

SELECT DISTINCT codigotit

FROM copia

WHERE costovolumen> (SELECT AVG (costovolumen)

FROM copia);



En esta imagen podemos ver que nuestro código funciona correctamente, ya que nos arroja los códigos de los libros que tienen un costo mayor al promedio total.

**8. Usando subconsultas, obtenga la información de los estudiantes de ingeniería civil que han prestado todos los libros del autor AT1.**

**PENDIENTE!!!!!**

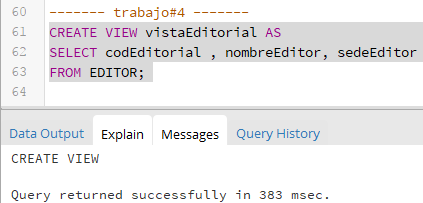
**Trabajo #4 (6 puntos)**

**Digite la instrucción:**

**CREATE VIEW vistaEditorial AS**

**SELECT codEditorial , nombreEditor, sedeEditor**

**FROM EDITOR;**

****

En esta imagen podemos ver el código en funcionamiento y el mensaje de confirmación de la creación de la vista.

**9. Ahora realice una consulta sobre la vista que se acaba de crear. Muestre en su informe el resultado. [2]**

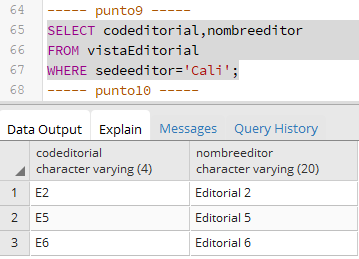
Para realizar esta consulta, quisimos mostrar el código de la editorial y el nombre del editor, de los que tienen sede en Cali.

Para ello utilizamos el siguiente código:

SELECT codeditorial,nombreeditor

FROM vistaEditorial

WHERE sedeeditor='Cali';



En esta imagen podemos ver que el código funciona perfectamente.

**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]**

Para la creación de esta vista utilizamos el código:

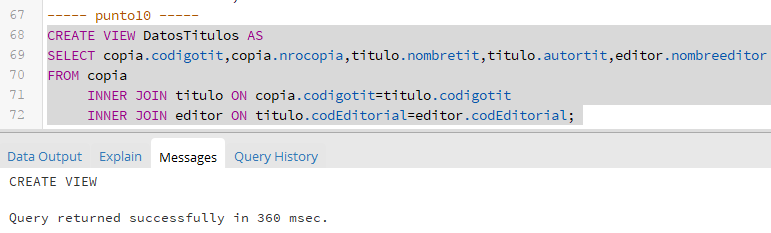
CREATE VIEW DatosTitulos AS

SELECT copia.codigotit,copia.nrocopia,titulo.nombretit,titulo.autortit,editor.nombreeditor

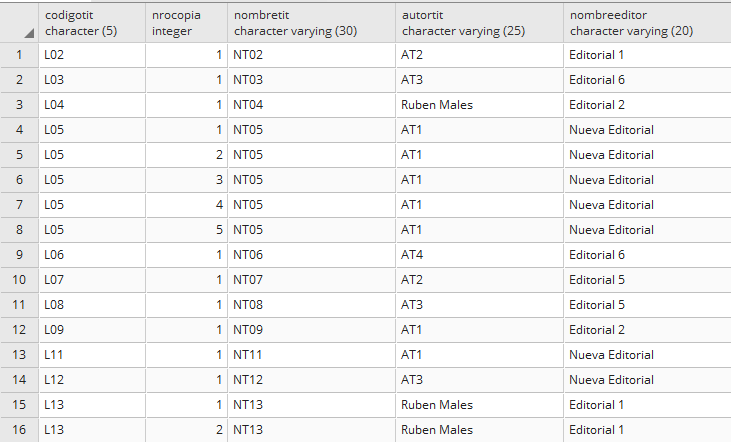
FROM copia

INNER JOIN titulo ON copia.codigotit=titulo.codigotit

INNER JOIN editor ON titulo.codEditorial=editor.codEditorial;

****

En la imagen anterior podemos ver el código utilizado, y observamos que nos arroja el mensaje de que la vista ha sido creada.





En la esta imagen podemos ver el resultado de nuestra vista, observamos que nos muestra cada uno de los atributos pedidos (código, numero de copia, nombre de la obra, autor y nombre del editor) de las obras que están en ambas tablas (titulo y copia). Note que la obra con código L01, está en la tabla copia pero no en la de titulo, por tanto, no nos lo muestra, ya que no hay información para todos los atributos.

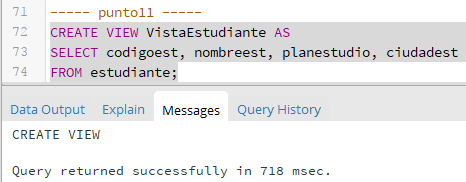
**11. Cree una vista de los estudiantes que contenga los datos del estudiante sin la dirección ni teléfono de cada estudiante. Asígnele el nombre VistaEstudiante como nombre a la vista. [2]**

Para crear tal vista, se utilizó el siguiente código:

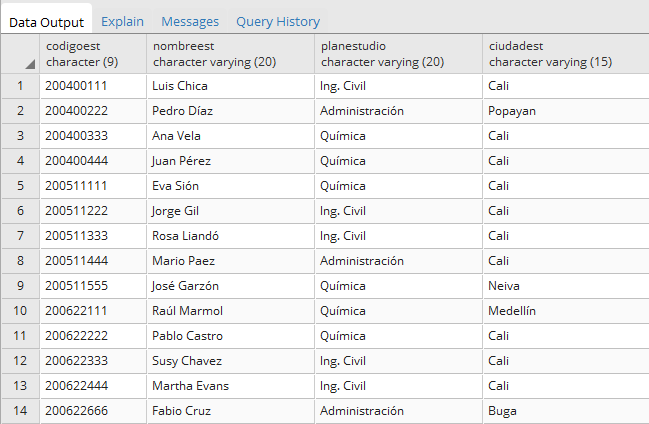
CREATE VIEW VistaEstudiante AS

SELECT codigoest, nombreest, planestudio, ciudadest

FROM estudiante;



En esta imagen podemos ver el código en funcionamiento y el respectivo mensaje de confirmación que nos dice que la vista ha sido creada.



En esta imagen podemos ver la vista denominada VistaEstudiante.

**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.**