

PROYECTO base de datos

MIGUEL ANGEL POVEA



13 Diciembre

2021

# Proyecto Base de Datos

Contenido

[Proyecto integrado Base de Datos 1](#_Toc90864466)

[INTRODUCCION 2](#_Toc90864467)

[ESTRUCTURA 3](#_Toc90864468)

[CREACION DE TABLAS 4](#_Toc90864469)

[TABLA MOTOS: 4](#_Toc90864470)

[TABLA CLIENTES 5](#_Toc90864471)

[TABLA VENDEDORES 5](#_Toc90864472)

[TABLA CONCESIONARIO 6](#_Toc90864473)

[TABLA MARCA 6](#_Toc90864474)

[INSERCION DE DATOS 6](#_Toc90864475)

[TABLA CLIENTES 6](#_Toc90864476)

[TABLA CONCESIONARIO 7](#_Toc90864477)

[TABLA MARCAS 8](#_Toc90864478)

[TABLA MOTOS 9](#_Toc90864479)

[TABLA VENDEDORES. 10](#_Toc90864480)

[CONSULTAS SQL 11](#_Toc90864481)

[CREACION DE ROLES, PRIVILEGIOS, USUARIOS. 11](#_Toc90864482)

[CREACION DE FUNCION Y PROCEDIMIENTO 13](#_Toc90864483)

[PROCEDIMIENTO 13](#_Toc90864484)

[FUNCION 14](#_Toc90864485)

[SEGURIDAD EN UNA BASE DE DATOS 15](#_Toc90864486)

[Copias de seguridad en frio 15](#_Toc90864487)

[Copias de seguridad en caliente 15](#_Toc90864488)

[Archivos REDOLOG 16](#_Toc90864489)

[Archivelog o noarchivelog, métodos de trabajo de una base de datos. 16](#_Toc90864490)

[Importación y Exportación de una base de datos. 17](#_Toc90864491)

[Exportación de la base de datos. 17](#_Toc90864492)

[Importación de la base de datos. 19](#_Toc90864493)

[Monitoreo de nuestra base de datos 21](#_Toc90864494)

[Optimización de la Base de Datos. 22](#_Toc90864495)

[RENDIMIENTO 22](#_Toc90864496)

[INSTANCIA 22](#_Toc90864497)

[PARTICIONAMIENTO Y PARALELIZACION 23](#_Toc90864498)

[CONSULTAS 24](#_Toc90864499)

[ALERTAS Y RECURSOS 24](#_Toc90864500)

[Enlaces de interés, información y anexo. 25](#_Toc90864501)

[Archive Logs. 26](#_Toc90864502)

[Seguridad. 26](#_Toc90864503)

[Rendimiento. 26](#_Toc90864504)

# INTRODUCCION

Vamos a crear una base de datos para el registro de un concesionario de motocicletas. Para almacenar las motos en la base de datos hay que, darle un código identificativo (la matricula), tenemos que saber en qué concesionario se encuentra, la marca a la que pertenece, sus kilómetros, el modelo, la antigüedad y el precio de la misma.

También interesa guardar datos de los concesionarios, en un concesionario de una ciudad solo puede haber una moto en concreto pero una misma moto solo podrá estar en un concesionario ya que las motos se mandarán a un concesionario u otro según la demanda que exista. Además de los concesionarios interesa guardar el nombre y la ciudad en la que se encuentre.

Cada moto pertenecerá a una misma marca, y a una marca pueden pertenecer varias motos.

Respecto a los concesionarios, en un concesionario habrá 1 o varias motos pero una moto en concreto solo puede encontrarse en un concesionario

Los clientes dependen de las motos que haya en el concesionario ya que si no hay motos en un concesionario no puede haber clientes. Hay que guardar Dni,Nombre,Apellidos,Localidad y Edad.

También hay que almacenar las distintas marcas de motos (Honda,Suzuki,Yamaha,Benelli) de las distintas motos cada marca tendrá un código identificativo y nombre. Una moto solo pertenece a una marca

Cada venta de moto tendrá un código identificativo. Hay que almacenar la fecha, precio de la venta, Matricula de la moto que se vende, El tipo de venta si se realiza en efectivo o con tarjeta así como el Dni del comprador. Cada venta tiene un id

Además cada marca paga a unos vendedores para que patrocinen las motos de sus marcas, un vendedor solo puede pertenecer a una marca, pero una misma marca puede tener más de un vendedor que le patrocine. De los vendedores debemos almacenar un identificativo, además del concesionario al que pertenecen. Un nombre, edad y sexo

# ESTRUCTURA

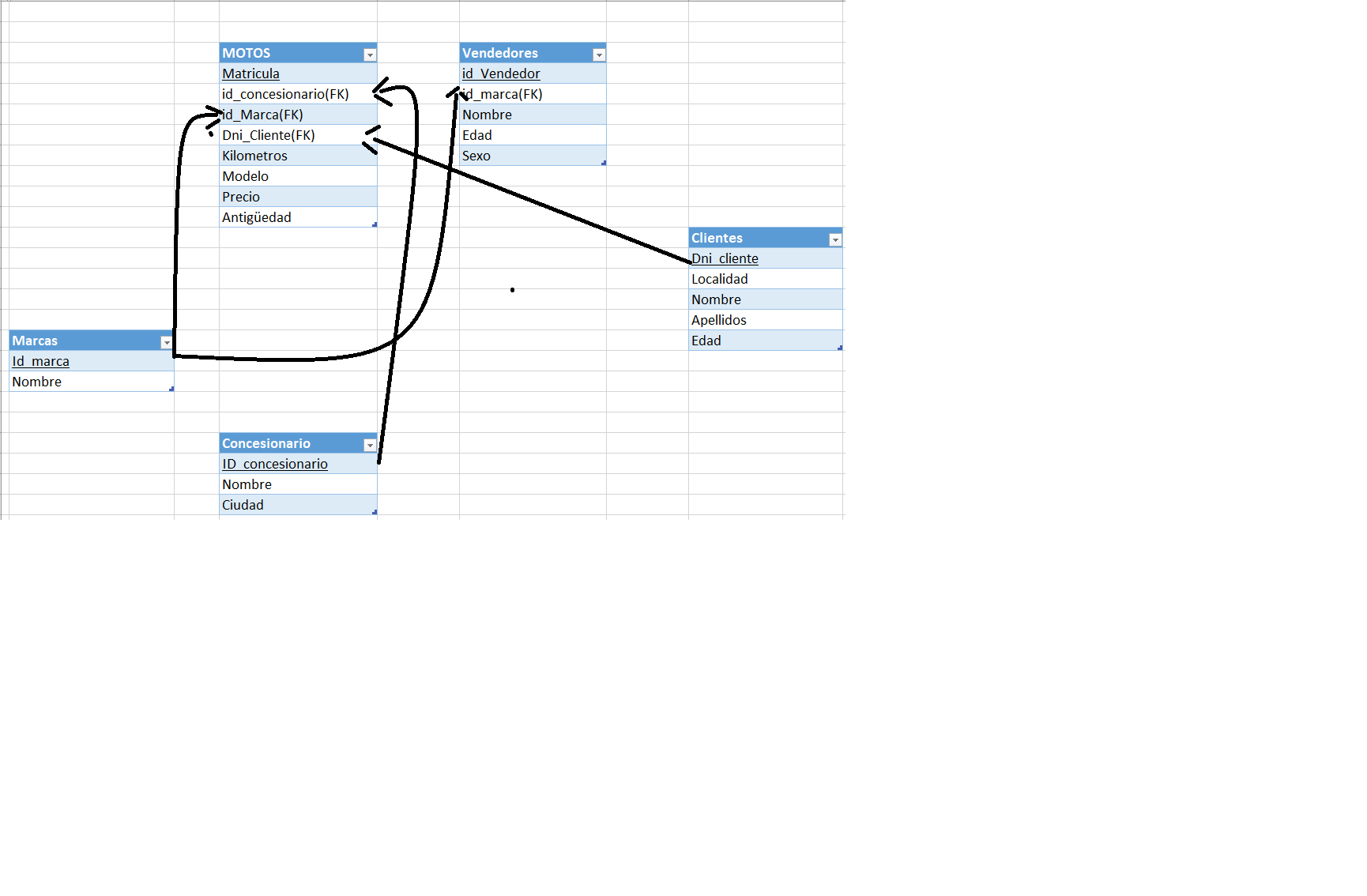
Motos( Matricula,id\_concesionario(FK),id\_Marca(FK),id\_cliente(FK),antigüedad,kilómetros,modelo,precio)

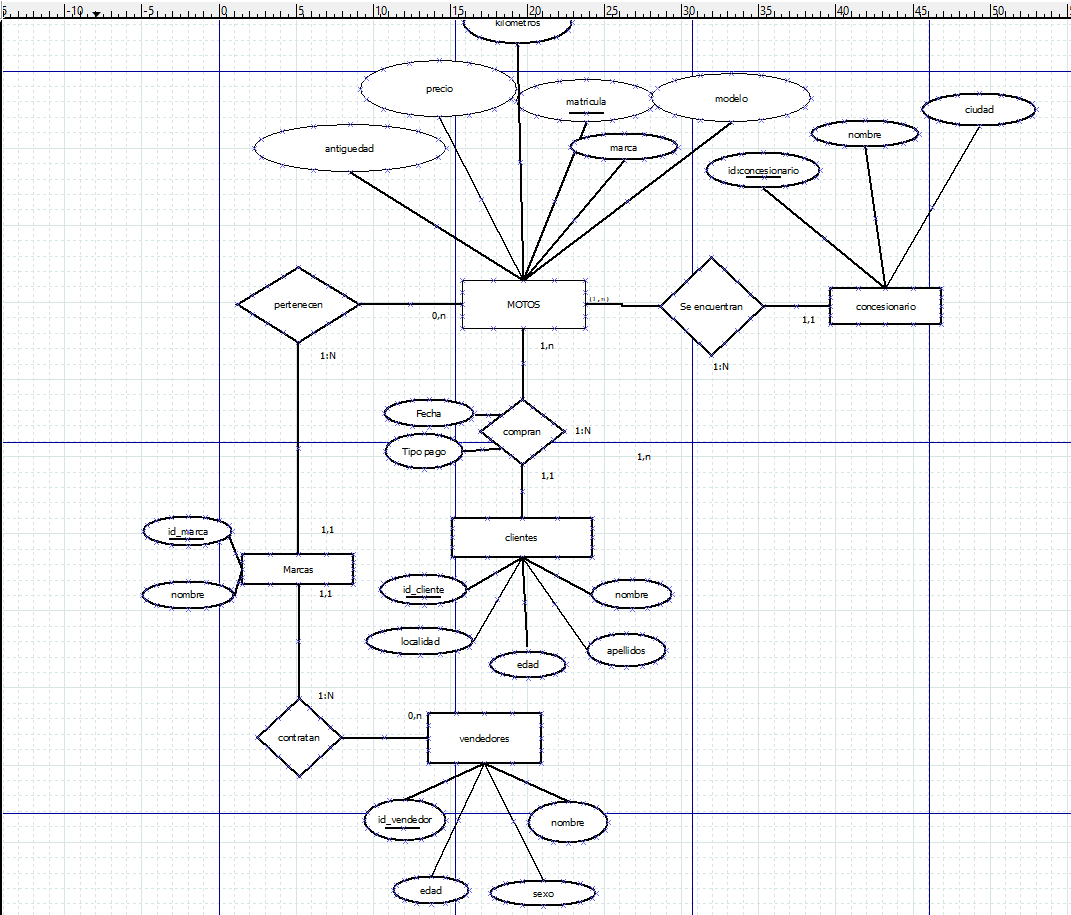
Concesionario(Id\_Concesionario,Nombre,Ciudad)

Marcas(id\_marca,Nombre)

Vendedores(ID\_Vendedor,Id\_marca (FK),Nombre,Edad,Sexo)

Clientes(Dni,Localidad,Nombre,Apellidos,Edad)





# CREACION DE TABLAS

LAS TABLAS QUEDARIAN ASI: (Estan creadas en el orden correcto aunque aquí en el pdf lo escriba en distinto orden)

## TABLA MOTOS:

CREATE TABLE MOTOS(

Matricula char(8) primary key,

Id\_Concesionario number not null,

Id\_Marca number not null,

Dni\_Cliente char(9) not null,

Marca varchar(20) not null,

Kilometros number not null,

Modelo varchar2(20) not null,

Precio number not null,

Antiguedad number not null,

CONSTRAINT FK\_Concesionarios foreign key(Id\_Concesionario) references

CONCESIONARIO (Id\_Concesionario),

CONSTRAINT FK\_Marcas foreign key(Id\_Marca) references MARCAS(Id\_Marca),

CONSTRAINT FK\_Cliente foreign key(Dni\_Cliente) references CLIENTES(Dni\_Cliente)

);

## TABLA CLIENTES

CREATE TABLE CLIENTES(

Dni\_Cliente char(9) primary key,

Localidad varchar2(20) not null,

Nombre varchar2(20) not null,

Apellidos varchar2(20) not null,

Edad number not null

);

## TABLA VENDEDORES

CREATE TABLE VENDEDORES(

Id\_Vendedor number primary key,

Id\_Marca number not null,

Nombre varchar2(20) not null,

Edad number not null,

Sexo varchar2(3) not null,

CONSTRAINT FK\_Marca foreign key(Id\_Marca) references

MARCAS (Id\_Marca)

);

## TABLA CONCESIONARIO

CREATE TABLE CONCESIONARIO(

Id\_Concesionario number primary key,

Nombre varchar2(20) not null,

Ciudad varchar2(20) not null

);

## TABLA MARCA

CREATE TABLE MARCAS(

Id\_Marca number primary key,

Nombre varchar2(20) not null

);

# INSERCION DE DATOS

## TABLA CLIENTES

INSERT INTO CLIENTES

(dni\_cliente, nombre, apellidos, edad, localidad)

VALUES('4922685E','PEPE','RODRIGUEZ','26','MAIRENA');

INSERT INTO CLIENTES

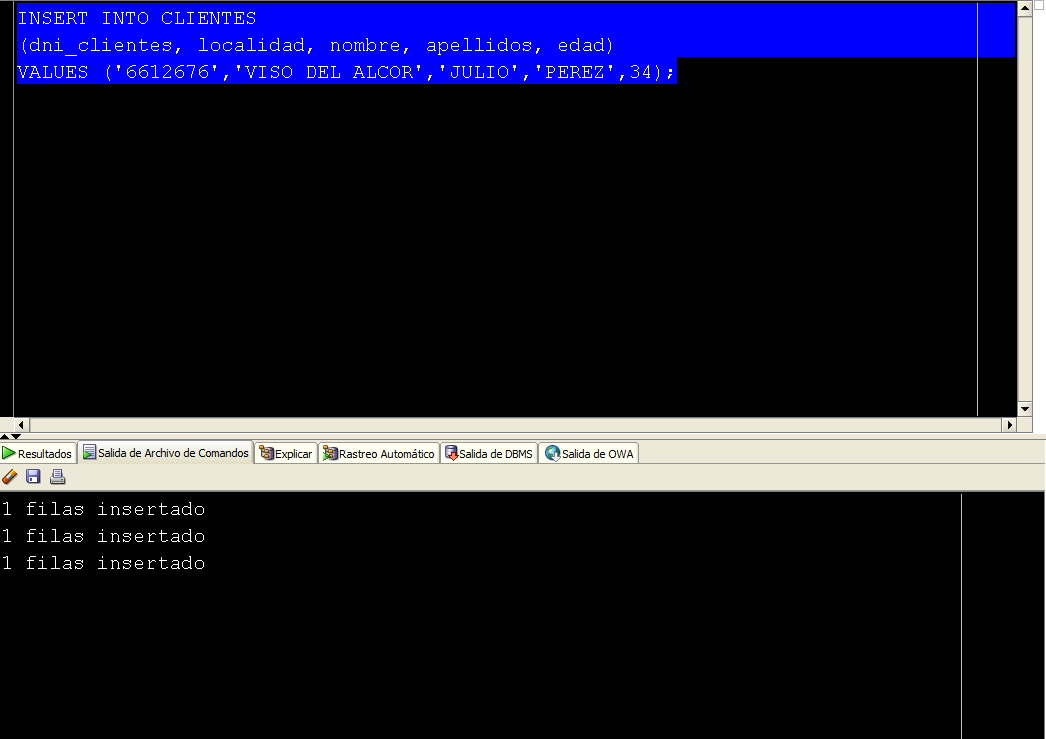
(dni\_cliente, nombre, apellidos, edad, localidad)

VALUES('911021E','MIGUEL','POVEA','22','VISO DEL ALCOR');

INSERT INTO CLIENTES

(dni\_cliente, nombre, apellidos, edad, localidad)

VALUES('123485E','ANA','INIESTA','56','MARCHENA');



## TABLA CONCESIONARIO

INSERT INTO CONCESIONARIO

(id\_concesionario, nombre, ciudad)

VALUES('22114','VELA S.L','SEVILLA ESTE');

INSERT INTO CONCESIONARIO

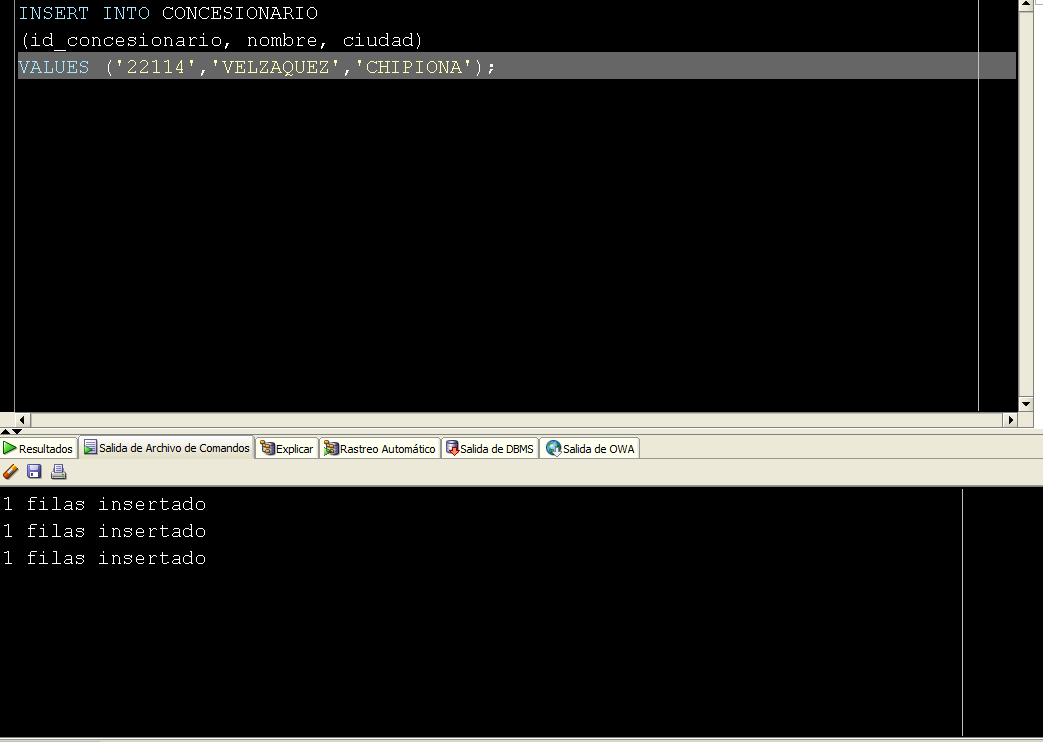
(id\_concesionario, nombre, ciudad)

VALUES('00002','LA TORRE','GELVES');

INSERT INTO CONCESIONARIO

(id\_concesionario, nombre, ciudad)

VALUES('61221','SPORT1000','ARAHAL');



## TABLA MARCAS

INSERT INTO MARCAS

(id\_marca, nombre)

VALUES('0','SUZUKI');

INSERT INTO MARCAS

(id\_marca, nombre)

VALUES('1','HONDA');

INSERT INTO MARCAS

(id\_marca, nombre)

VALUES('2','YAMAHA');

INSERT INTO MARCAS

(id\_marca, nombre)

VALUES('3','BMW');

INSERT INTO MARCAS

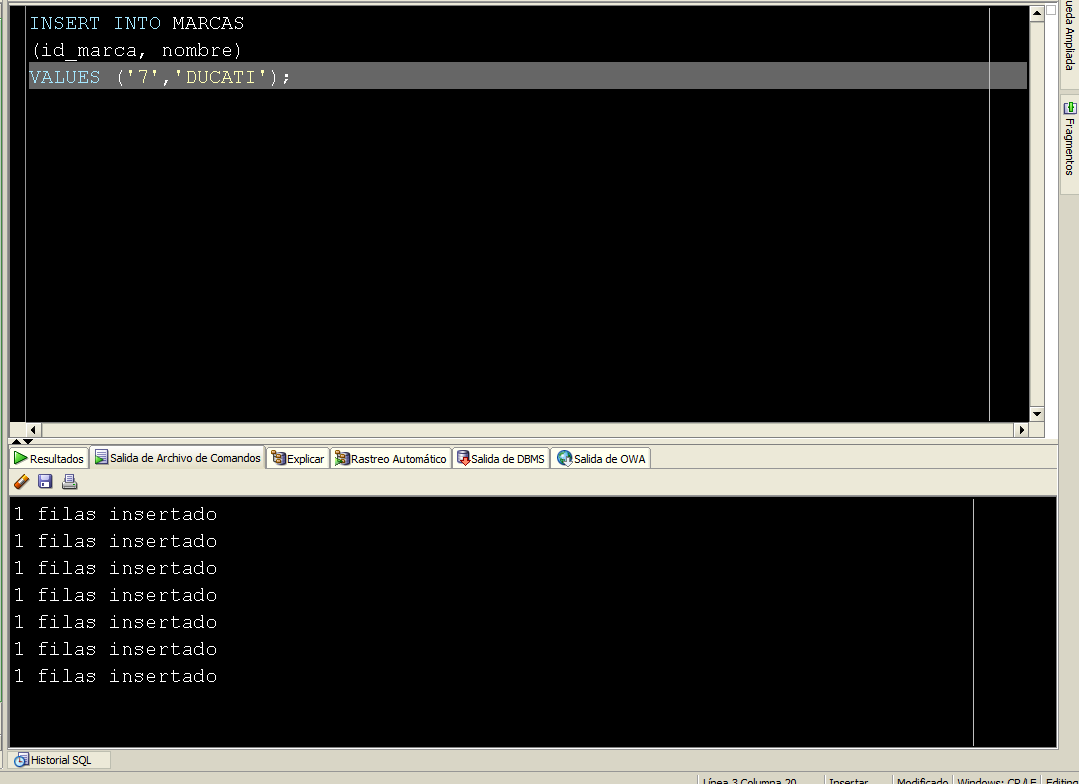
(id\_marca, nombre)

VALUES('4','DUCATI');

INSERT INTO MARCAS

(id\_marca, nombre)

VALUES('5','HYOSUNG');



## TABLA MOTOS

INSERT INTO MOTOS

(matricula, id\_concesionario, id\_marca, dni\_cliente, marca, modelo, kilometros, precio, antiguedad)

VALUES('3453ABC','2','0','4922685E','SUZUKI','GSX650F','54000','2700','2007');

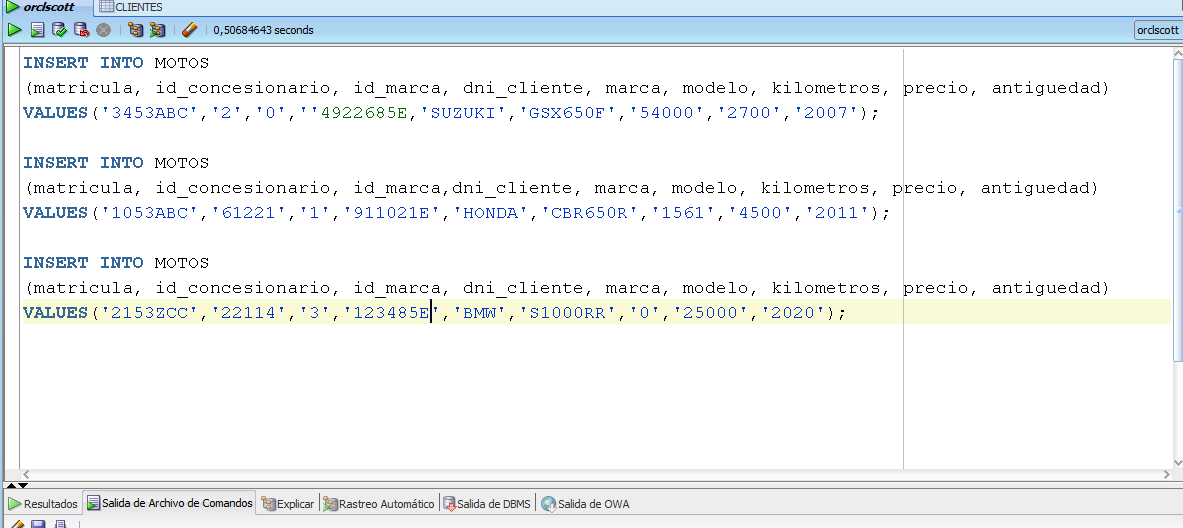
INSERT INTO MOTOS

(matricula, id\_concesionario, id\_marca,dni\_cliente, marca, modelo, kilometros, precio, antiguedad)

VALUES('1053ABC','61221','1','911021E','HONDA','CBR650R','1561','4500','2011');

INSERT INTO MOTOS

(matricula, id\_concesionario, id\_marca, dni\_cliente, marca, modelo, kilometros, precio, antiguedad)

VALUES('2153ZCC','22114','3','123485E','BMW','S1000RR','0','25000','2020');

## TABLA VENDEDORES.

INSERT INTO VENDEDORES

(id\_vendedor, id\_marca, nombre, edad, sexo)

VALUES('66','1','PEPE','18','M');

INSERT INTO VENDEDORES

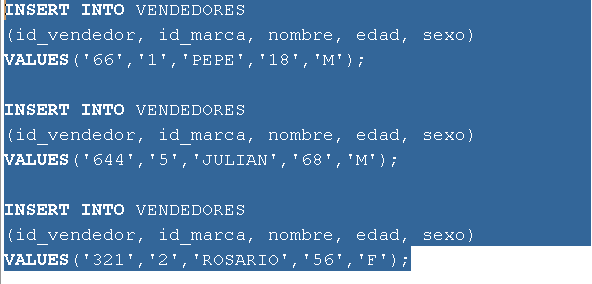
(id\_vendedor, id\_marca, nombre, edad, sexo)

VALUES('644','5','JULIAN','68','M');

INSERT INTO VENDEDORES

(id\_vendedor, id\_marca, nombre, edad, sexo)

VALUES('321','2','ROSARIO','56','F');



# CONSULTAS SQL

1. OBTENER EL MODELO DE TODAS LAS MOTOS QUE CUESTEN MAS DE 3000€

select modelo from motos where precio > '3000';

1. OBTEN LOS KILOMETROS, LA MARCA Y EL PRECIO DE LAS MOTOS CUYA ANTIGUEDAD ES IGUAL AL AÑO 2007

select kilometros, marca, precio from motos where antiguedad = '2007';

1. Mostrar todos los datos de la moto cuyo modelo sea “GSX650F”.

Select \* from motos where modelo = ‘GSX650F’;

1. OBTENER EL NOMBRE DE LA MARCA Y NUMERO DE VENDEDOR, DE AQUEL VENDEDOR CUYO NOMBRE ACABE EN “E” Y SU EDAD ESTÉ COMPRENDIDA ENTRE 18 Y 20 AÑOS
2. Select m.nombre AS LA\_MARCA,v.nombre AS EL\_NOMBRE\_VENDEDOR from marcas m,vendedores v where m.id\_marca=v.id\_marca and v.nombre LIKE '%E' and edad in (18,20);

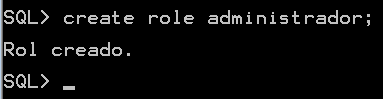
# CREACION DE ROLES, PRIVILEGIOS, USUARIOS.

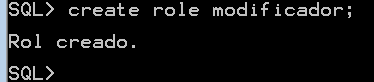
1 CREACION DE USUARIOS



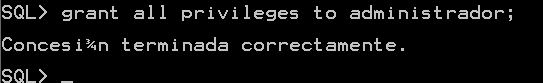


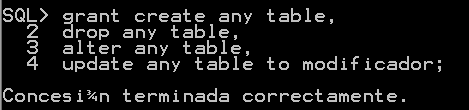
2 CREACION DE ROLES



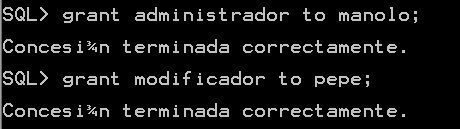


3 ADMINISTRACION DE PRIVILEGIOS DE LOS ROLES

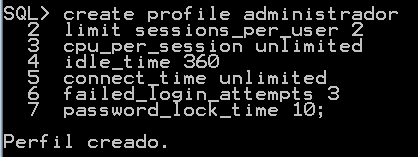


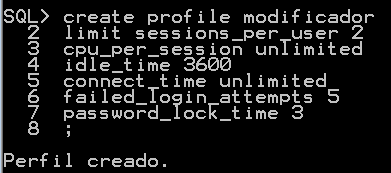


4 ASIGNACION DE ROLES A USUARIOS

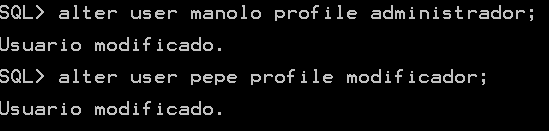


5 CREACION DE PERFILES





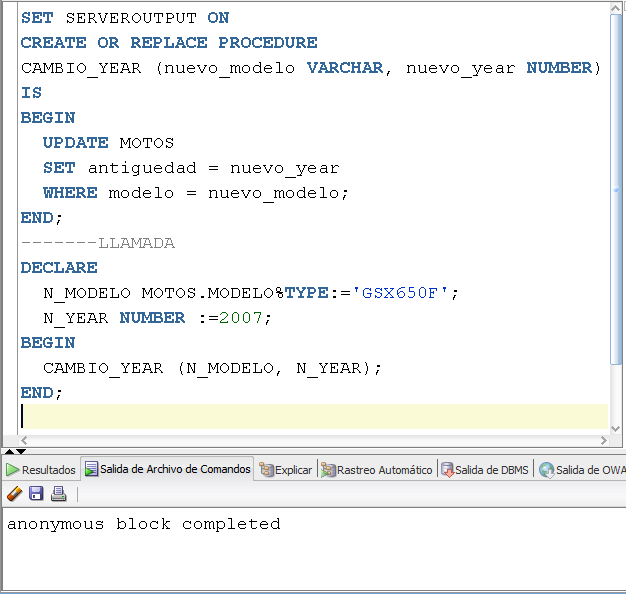
6 ASIGNACION DE PERFILES A USUARIOS



# CREACION DE FUNCION Y PROCEDIMIENTO

## PROCEDIMIENTO

CREAREMOS UN PROCEDIMIENTO QUE INTRODUCIENDO EL MODELO DE LA MOTO NOS DEJE CAMBIAR EL AÑO EN EL QUE FUE FABRICADA.

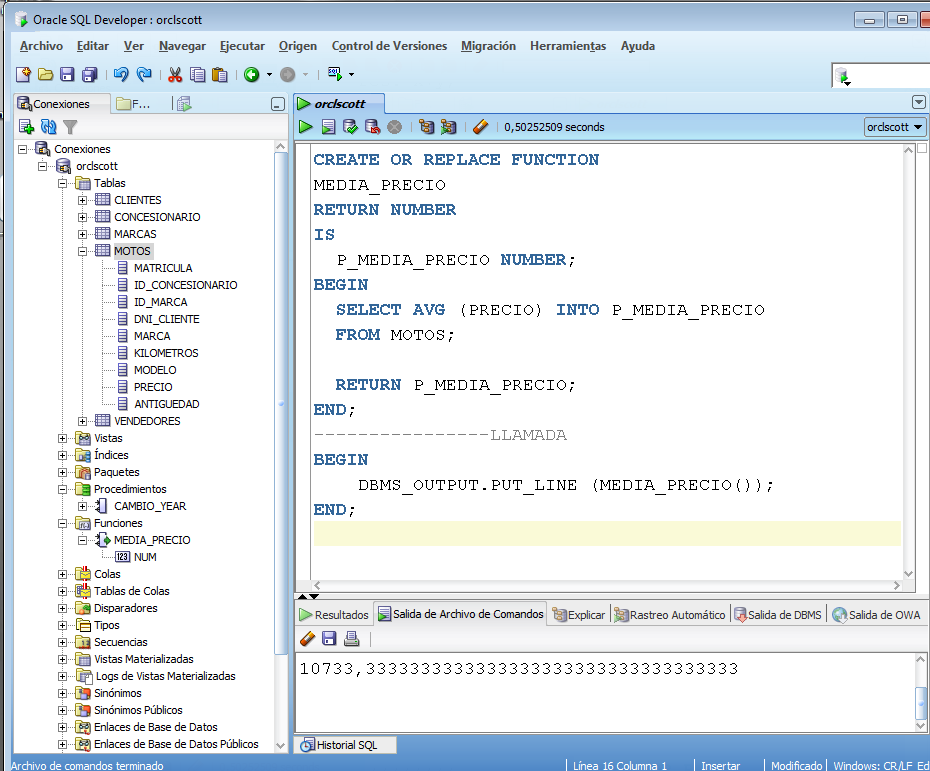




Comprobamos que lo cambia.

## FUNCION

VAMOS A REALIZAR UNA FUNCION QUE NOS PERMITA OBTENER LA MEDIA DE PRECIOS DE MOTOCICLETAS, AL LLAMAR A LA FUNCION NOS DEBE APARECER LA MEDIA POR PANTALLA.



# SEGURIDAD EN UNA BASE DE DATOS

Las copias de seguridad de una base de datos garantiza el acceso a los datos ante todo tipo de problemas tales como problemas de hardware del propio servidor, virus o amenazas, vulnerabilidades y así como ataques ransomware o secuestro de datos.

Es conveniente realizar copias de seguridad de nuestra base de datos para así evitar fácilmente esta serie de problemas. Por ello, con el fin de evitarlos, surgen dos tipos de copias de seguridad.

## Copias de seguridad en frio

Este tipo de copias de seguridad se caracteriza por ser realizada con la base de datos totalmente desconectada, así como los servicios que estén relacionados con nuestra Base de Datos.

Este tipo de copias de seguridad tiene una serie de ventajas e inconvenientes entre los cuales se encuentran:

La rapidez de este tipo de copias de seguridad frente a los backup lógicos.

Una reducción en la probabilidad de inconsistencias.

Por contraparte, también presenta como una muy gran desventaja, el hecho de que tengamos que renunciar a nuestra base de datos de forma temporal, así como de los servicios relacionados a ella.

A continuación mostraré la creación de una copia física en frío.

En primer lugar debemos copiar una serie de ficheros:

## Copias de seguridad en caliente

En este otro tipo de copias de seguridad, a diferencia de la anterior, la base de datos se encuentra activa, asi como los servicios de la misma.

Las ventajas de este tipo de backup son las siguientes:

-Mayor rapidez frente a los backups de tipo lógicos.

-La base de datos está disponible en todo momento.

Sus desventajas:

-Existe una alta probabilidad de que nuestra BD sufra inconsistencias.

A continuación, mostrare como realizar una copia de seguridad lógica en caliente:

En primer lugar en nuestra terminal de SQL Plus debemos comprobar que el módulo mysqldump se encuentra instalado, para ello podemos hacer uso de la sentencia:

Mysqldump –version

Además de esto, debemos de tener en cuenta que el usuario con el que estemos conectado debe disponer de permisos para realizar la función select sobre la base de datos.

Dicho esto, si queremos proceder con la creación de una copia de seguridad debemos de ejecutar el siguiente comando:

Mysqldump –u (usuario) –p (nombre de nuestra BBDD) **> (nombre del fichero donde queremos guardarlo con la extensión.sql)**

**Como vemos, la sintaxis es muy simple, un ejemplo de copia de seguridad podría ser el siguiente:**

**Mysqldump –u system –p concesionario > backup.sql**

**Y ya tendríamos la base de datos exportada a un fichero .sql**

**Si lo que queremos es la importación de nuestra base de datos, el proceso sería el mismo, únicamente debemos modificar el signo “Mayor que” por el signo “Menor que”.**

## **Archivos REDOLOG**

Este tipo de archivos son pertenecientes a nuestra base de datos. En ellos se almacenan los cambios que ocurren en nuestra base de datos. Gracias a estos datos, ante cualquier inconveniente podremos disponer de un registro de todos los movimientos de la base de datos.

Normalmente son dos los archivos redolog, uno es el encargado de recoger los datos que se extraen de la base de datos mientras que el otro los almacena en nuestra unidad primaria.

## Archivelog o noarchivelog, métodos de trabajo de una base de datos.

Antes de comenzar, vamos a indicar cual es la forma de comprobar en cual de estos 2 estados se encuentra nuestra base de datos y es que con el uso del siguiente comando podremos averiguarlo con total facilidad:

Archive log list

Nos indicará el modo log de nuestra BD (Modo de archivado o Modo de no archivado).

La forma de trabajar de nuestra base de datos puede encontrarse únicamente en uno de estos dos modos distintos de almacenamiento de logs.

En el modo archivelog, los datos se almacenan en unos archivos denominados logs en los cuales se guarda la información necesaria para poder llevar la base de datos a un punto concreto de su vida útil. Gracias a este modo, la base de datos puede recuperar archivos corruptos mediante los logs, sin la necesidad de detener la BBDD.

En cambio, en el modo no archivelog renunciamos a lo dicho anteriormente, puesto que no podremos recuperar el estado de la base de datos en un momento concreto. Además de esto, con este modo de trabajo las copias de seguridad no pueden ser de tipo online.

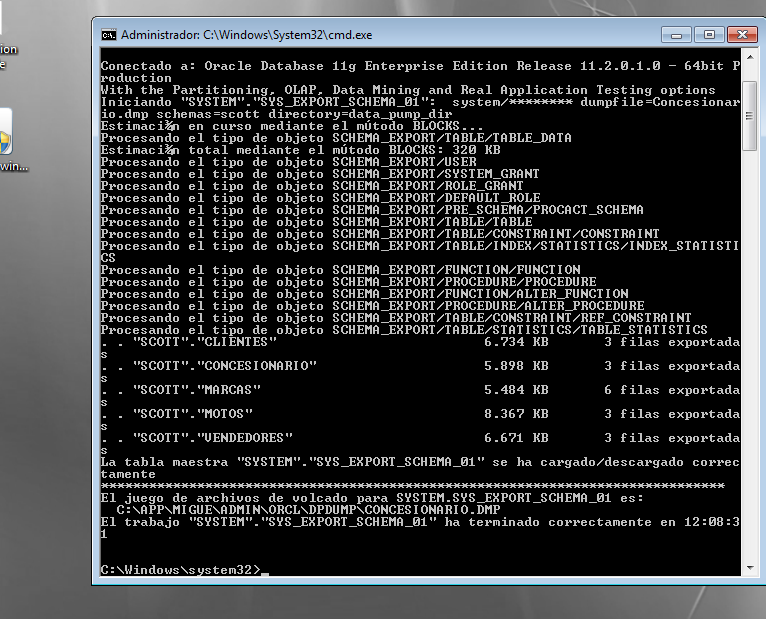
# Importación y Exportación de una base de datos.

## Exportación de la base de datos.

Para la exportación de nuestra base de datos, debemos de seguir los siguientes pasos.

Lo primero será ejecutar la consola de sql e introducir los siguientes comandos.

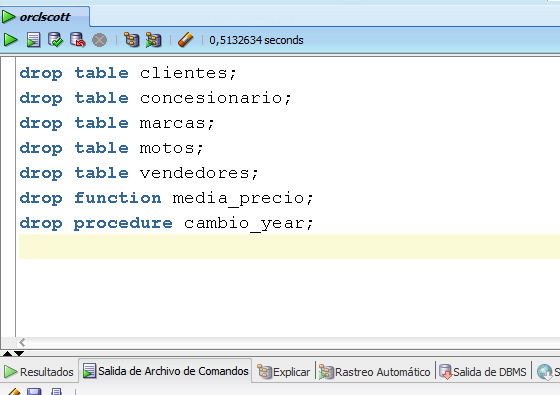
expdp system/admin11G dumpfile=Concesionario.dmp schemas=scott directory=data\_pump\_dir



Así exportaremos toda la base de datos, en cambio si deseamos exportar una única tabla lo haríamos de la siguiente forma.

expdp system/admin11G dumpfile=MOTOS.DMP tables=SCOTT.MOTOS directory=data\_pump\_dir

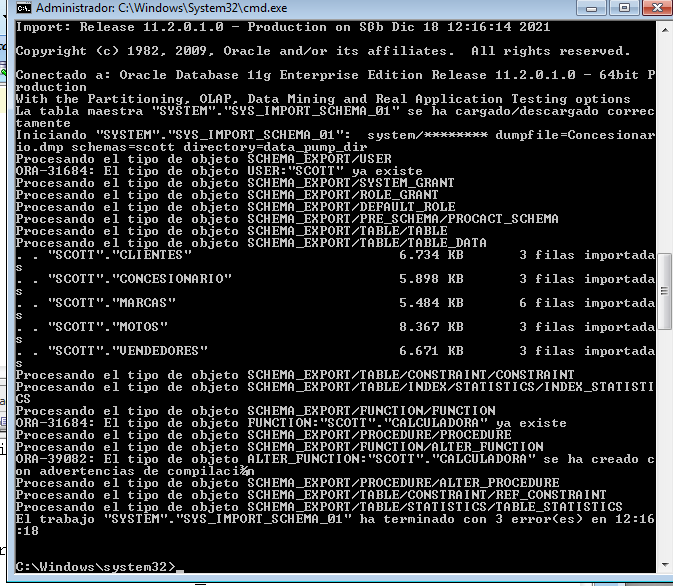
Hecho esto, ya podemos borrar las tablas con total tranquilidad.



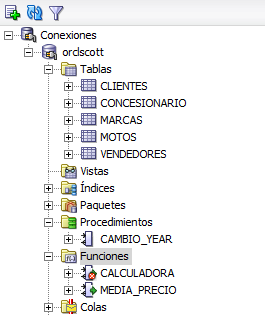
# Importación de la base de datos.

Solo si hemos realizado los pasos anteriores podremos proceder con la importación de nuestra base de datos. Para ello debemos de introducir el siguiente comando.

impdp system/admin11G dumpfile=Concesionario.dmp schemas=scott directory=data\_pump\_dir



Y nos aparecen nuevamente las tablas, funciones y procedimientos.



# Monitoreo de nuestra base de datos

Para el monitoreo de nuestra base de datos existe una herramienta, llamada AWR.

Esta nos permite:

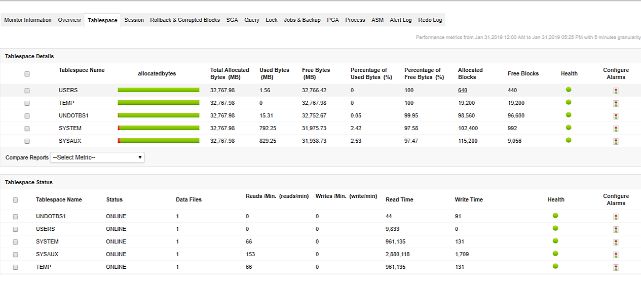
▪ Medir los tiempos de respuesta cada hora.

▪ Monitorear las mediciones de rendimiento.

▪ Identificar cuellos de botella en el rendimiento, así como notificar a los administradores de la base de datos de los mismos.

▪ Realizar analíticas de rendimiento de nuestra BBDD.

Dicha herramienta dispone de una interfaz gráfica intuitiva para el usuario mediante la cual se obtiene información sobre la carga del servidor.



Vistas y funciones dinámicas.

Estas devuelven datos sobre el estado del servidor y pueden ayudar a la hora de diagnosticar problemas y optimizar el rendimiento.

Sys.dm\_os\_vulome\_stats (Permite obtener datos sobre archivos físicos que componen el almacenamiento de nuestra BD)

Sys.dm\_server\_services(Nos sirve para acceder al listado de servicios relacionados con SQL Server así como su nombre, cuenta de arranque y estado)

Sys.dm\_os\_windows\_info(Sirve para conocer la versión del SO de la instancia SQLServer)

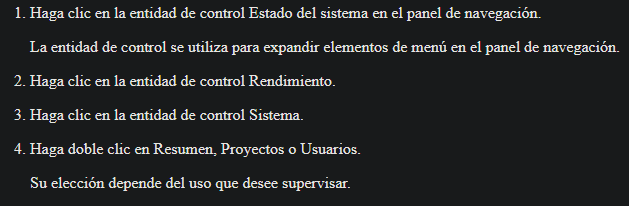
Además de éstas, existen infinidad de funciones para administrar nuestro servidor.

# Optimización de la Base de Datos.

## RENDIMIENTO

Dentro de Oracle, también existe una herramienta de rendimiento a la cual podemos acceder.

Para acceder a ella:



En ella podemos encontrar las siguientes utilidades.



## INSTANCIA

Si queremos mejorar el rendimiento de la instancia de nuestra BBDD tenemos que tener en cuenta el hardware de nuestra máquina, puede que el aumento de algo tan simple como la memoria RAM nos parezca una minucia, pero un aumento en la memoria RAM de nuestra instancia tendrá una gran repercusión en el rendimiento de la misma.

En las instancias SQL Server la memoria RAM puede limitarse, de hecho, es muy recomendable hacerlo, ya que si ignoramos esta recomendación SQL Server actúa consumiendo insaciablemente toda nuestra memoria principal, pudiendo afectar al rendimiento de otros servicios.

Otra recomendación es el reconstruir índices, aunque no notaremos una gran mejora en bases de datos en la cual se reconstruyen semanalmente, en una BD que nunca haya sido ejecutado el script para recuperar índices, notaremos una mejora de rendimiento bastante considerable.

## PARTICIONAMIENTO Y PARALELIZACION

Para realizar particiones en MYSQL…

Una vez tenemos una tabla creada debemos de ejecutar esta sentencia que en este caso vamos a particionar una serie de informes ordenándolos por mes

ALTER TABLE reports PARTITION BY RANGE(TO\_DAYS(date))(

PARTITION p201111 VALUES LESS THAN (TO\_DAYS("2011-12-01")),

PARTITION p201112 VALUES LESS THAN (TO\_DAYS("2012-01-01")),

PARTITION p201201 VALUES LESS THAN (TO\_DAYS("2012-02-01")),

PARTITION p201202 VALUES LESS THAN (TO\_DAYS("2012-03-01")),

PARTITION p201203 VALUES LESS THAN (TO\_DAYS("2012-04-01")),

PARTITION p201204 VALUES LESS THAN (TO\_DAYS("2012-05-01")),

PARTITION p201205 VALUES LESS THAN (TO\_DAYS("2012-06-01")),

PARTITION pDefault VALUES LESS THAN MAXVALUE

);

La ultima partición pDefault contara con todos los registros que no entren en las particiones anteriores, por lo que la información nunca dejara de insertarse en la tabla.

Para borrar particiones usaremos la siguiente sintaxis.

ALTER TABLE reports DROP PARTITION p201111;

Con esto borraríamos la primera partición.

Si queremos consultar información sobre particiones ya creadas podemos hacerlo con el siguiente comando

SELECT PARTITION\_NAME,TABLE\_ROWS FROM information\_schema.PARTITIONS WHERE TABLE\_NAME='reports';

Otras de las cosas a tener en cuenta a la hora de optimizar nuestra base de datos, son las consultas. Para ello debemos de centrarnos en los siguientes aspectos.

## CONSULTAS

Como primer aspecto a tener en cuenta, antes de crear nuestra base de datos debemos de asegurarnos que los tipos de datos que vamos a introducir sean lo más adecuados al valor que queremos representar para así no desaprovechar recursos.

Además, debemos de añadir al nombre de la tabla el propietario, para que el esquema haga más rápida la consulta.

Usar el nombre de las columnas en lugar del \*

Usar el not exists en lugar del not in.

Evitar unir tablas con where, en su lugar trabajar con con join.

Evitar las cláusulas distinct, group by y order by ya que consumen más memoria.

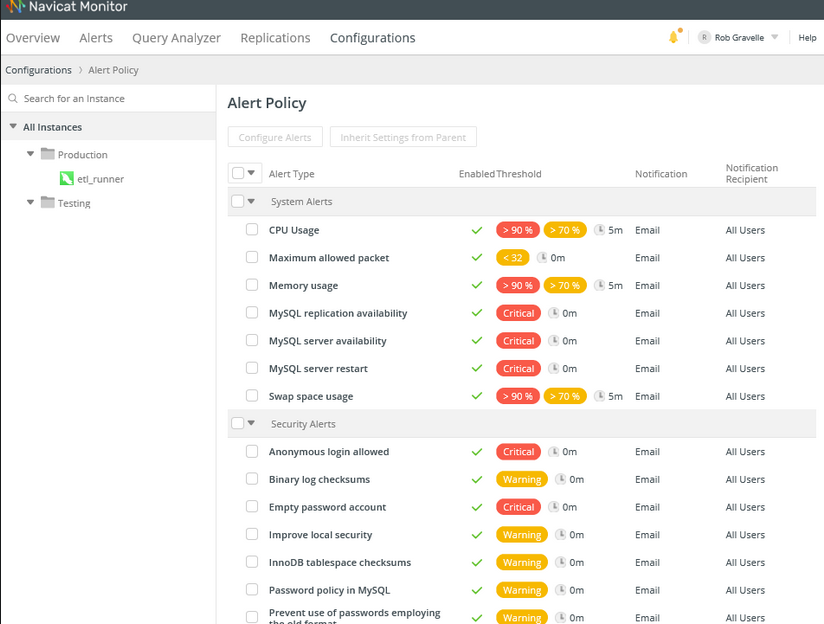
## ALERTAS Y RECURSOS

En MySQL existen unas herramientas llamadas alertas, estas permiten enviar notificaciones o que se realicen ciertas acciones en determinadas situaciones de nuestro SGBD.

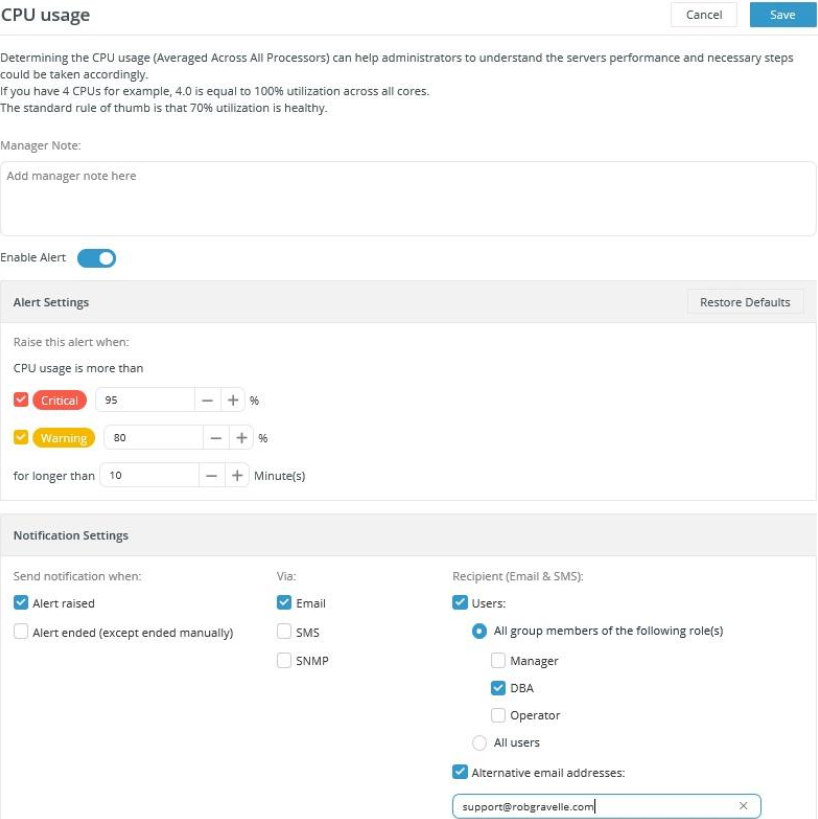
Existen diversas herramientas, entre ellas Navicat .

Navicat es una herramienta que permite el monitoreo de nuestra base de datos desde un navegador web, aunque no es lo más interesante de navicat.

Lo más interesante, es la capacidad de notificar por correo electrónico, SMS… cuando hay alertas en la infraestructura, además cuenta con alarmas personalizadas



En ellas se puede establecer políticas de alerta, por ejemplo, sobre el uso de CPU cuando se supere un % de uso de CPU determinado. En la misma aplicación podemos elegir si enviar un correo electrónico en el momento que se supere el 80% de uso de CPU ya sea al instante, 10 minutos más tarde…



Ésta es sin duda una gran aplicación a la hora de trabajar con una base de datos pues resulta de gran utilidad.

# Enlaces de interés, información y anexo.

### Archive Logs.

<https://orasite.com/tutoriales/instalacion/configurar-base-datos-modo-archivelog>

### Seguridad.

<https://www.ajpdsoft.com/modules.php?name=News&file=print&sid=559>

<https://asir2tasgbd.wordpress.com/2016/03/07/copias-de-seguridad-en-mysql/>

### Rendimiento.

<https://guidocutipa.blog.bo/optimizar-rendimiento-mariadb-mysql-parte-2/>

<https://ikastaroak.birt.eus/edu/argitalpen/backupa/20200331/1920k/es/ASIR/ASGBD/ASGBD04/es_ASIR_ASGBD04_Contenidos/website_14_herramientas_para_la_creacin_de_alertas_de_rendimiento.html>

### Navicat.

<https://programmerclick.com/article/3470368495/>

THE END