**TECNOLOGICO NACIONAL DE MEXICO CAMPUS CULIACAN INGENIERIA EN SISTEMAS COMPUTACIONALES**

Imagen que contiene Logotipo

Descripción generada automáticamente

**MATERIA**

ADMINISTRACION DE BASE DE DATOS

**INTEGRANTES**

LOZANO CORVERA DANIEL ANTONIO

GARCIA AGUILAR JOSE ALFREDO

**MAESTRO**

DANIEL ESPARZA SOTO

**FECHA**

26-SEPTIEMBRE-2022

**PROYECTO ORACLE**

1. **RESUMEN DE LA HISTORIA**

Oracle es un sistema de gestión de bases de datos que se utiliza en todo el mundo en varias empresas. Oracle hace uso de lenguaje SQL o ‘Sequel’ para la manipulación de datos y operaciones en datos. Hoy, en este artículo, revisaremos la base de datos History of Oracle.

Oracle surge en 1977 bajo el nombre de SDL (Software Development Laboratories).

En 1977, dos programadores informáticos llamados Larry Ellison & Bob Miner iniciaron Oracle Corporation. Tanto Larry como Bob tenían una buena experiencia en el desarrollo de programas de bases de datos para varias empresas. Y obtuvieron su primer proyecto que consistía en construir el programa de base de datos para la Agencia Central de Inteligencia o CIA.

En 1979, SDL cambia su nombre por Relational Software, Inc. (RSI). Trabajó en una provincia hasta 1982.

Larry, Bob & Ed Oates se centraron constantemente en los sistemas de gestión de bases de datos e hizo su prueba inicial de base de datos con la empresa IBM y fracasó. En estas pruebas posteriores, Oracle Republic System liderado por Bob fue un gran éxito. Entonces, el nombre de Oracle Corporation fue cambiado a Oracle Systems Corporation. En 1995, esto se cambió nuevamente a Oracle Corporation y este nombre continúa incluso el día.

Fue la primera base de datos diseñada solo para la computación en red empresarial y la forma más rentable y flexible de administrar aplicaciones e información.

1. **VERSIONES**

* 1978 (Oracle 1) – Nunca oficial
* 1979 (Oracle v2)
* 1983 (Oracle v3)
* 1984 (Oracle v4)
* 1985 (Oracle v5)
* 1988 (Oracle v6)
* 1988 (Oracle 6.2)
* Junio 1992 (Oracle 7)
* Mayo 1994 (Oracle 7.1)
* Mayo 1995 (Oracle 7.2)
* Febrero 1996 (Oracle 7.3)
* Junio 1997 (Oracle 8 Database)
* 1998 (Oracle 8i/Database)
* 2001 (Oracle 9/Database)
* 2002 (Oracle 9/Database Release 2)
* 2003 (Oracle Database 10g Release 1)
* Julio 2005 (Oracle Database 10g Release 2)
* Septiembre 2007 (Oracle Database 11g Release 1)
* Septiembre 2009 (Oracle Database 11g Release 2)
* Julio 2013 (Oracle Database 12c Release 1)
* Marzo 2017 (Oracle Database 12c Release 2)
* Julio 2018 (Oracle Database 18c)
* Abril 2019 (Oracle Database 19c)
* Agosto 2021 (Oracle Database 21c)

Solo las versiones subrayadas siguen teniendo soporte o mantenimiento. Las que no, están obsoletas.

Productos de Oracle:

* Enterprise Edition Este buque insignia de Oracle ofrece varias funciones y herramientas para las grandes corporaciones.
* Edición estándar: la edición estándar de Oracle tiene una función de administración de base de datos básica para las tiendas pequeñas y medianas a un costo menor que la edición empresarial.
* Standard Edition One: Oracle SEO tiene un precio único para un solo servidor de CPU utilizado por pequeñas empresas.
* Oracle Express: tiene algunas características de DBA que incluyen replicación de solo lectura.

1. **CARACTERÍSTICAS**

* Modelo relacional: los usuarios visualizan los datos en tablas con el formato filas/columnas.
* Herramienta de administración gráfica intuitiva y cómoda de utilizar.
* Control de acceso: tecnologías avanzadas para vigilar la entrada a los datos.
* Protección de datos: seguridad completa en el entorno de producción y de pruebas y gestión de copias de seguridad.
* Lenguaje de diseño de bases de datos muy completo (PL/SQL): permite implementar diseños "activos", que se pueden adaptar a las necesidades cambiantes de negocio.
* Alta disponibilidad: escalabilidad, protección y alto rendimiento para la actividad empresarial.
* Gestión de usuarios: agilidad en los trámites, reducción de costes y seguridad en el control de las personas que acceden a las aplicaciones y a los sistemas.
* La base de datos de Oracle se ejecuta en varios sistemas operativos. Esto se integra con aplicaciones y plataformas de terceros, por lo que esas aplicaciones están totalmente respaldadas por la base de datos.
* Proporciona una plataforma técnica para sus clientes. La base de datos de Oracle es compatible con ACID y proporciona copias de seguridad de datos.
* Oracle RAC permite a los usuarios ejecutar una sola base de datos en varios servidores. Es muy útil cuando el servidor experimenta una interrupción; esta base de datos puede moverse a una instancia diferente y continuar realizando operaciones allí, aunque otras han caído por completo.
* Disponibilidad en la nube
* Oracle hace uso de SQL como lenguaje de consulta para otras bases de datos
* Motor de base de datos objeto-relacional más usado a nivel mundial.
* Multiplataforma: puede ejecutarse desde un PC hasta una supercomputadora.
* Permite el uso de particiones para hacer consultas, informes, análisis de datos, etc.
* Soporta todas las funciones que se esperan de un buen servidor.
* Software del servidor que puede ejecutarse en multitud de sistemas operativos: Linux, Mac, Windows, etc.

1. **MANUAL DEL PROCESO DE INSTALACIÓN**

Buscamos descargar Oracle en el navegador de nuestra preferencia y le damos buscar:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Seleccionamos esta de la página oficial:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Nos saldrá esta pagina y le daremos al botón verde señalado:

Interfaz de usuario gráfica

Descripción generada automáticamente

Ahora te saldrá esta página para instalar, aquí dependerá des sistema operativo que tengas en este caso nosotros tenemos Windows de 64bits:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Una ves que le dimos clic se comenzara a descargar:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente con confianza media

Una vez que se descargue la extraemos:

Interfaz de usuario gráfica

Descripción generada automáticamente

Lugo buscamos el instalador y lo ejecutamos como administrador:



Se nos abrirá esta pestaña y le damos siguiente:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Después tenemos que marcar donde dice acepto y nuevamente darle aceptar:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Te mostrara en la carpeta que se va a guardar nuevamente le damos siguiente:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Luego de va a pedir una contraseña para la base de datos, la introducimos 1234 luego le damos a siguiente:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Comenzará la instalación:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Nos saldrá este mensaje y le daremos permitir:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Luego vamos al buscador de nuestra preferencia para descargar su interfaz gráfica:

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

Descripción generada automáticamente

Aquí descargaremos la que nos corresponde dependiendo de nuestro sistema operativo en nuestro caso es la que esta señalada:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Al querer descargarlo nos pedirá que creemos una cuenta de Oracle:

Captura de pantalla de un celular

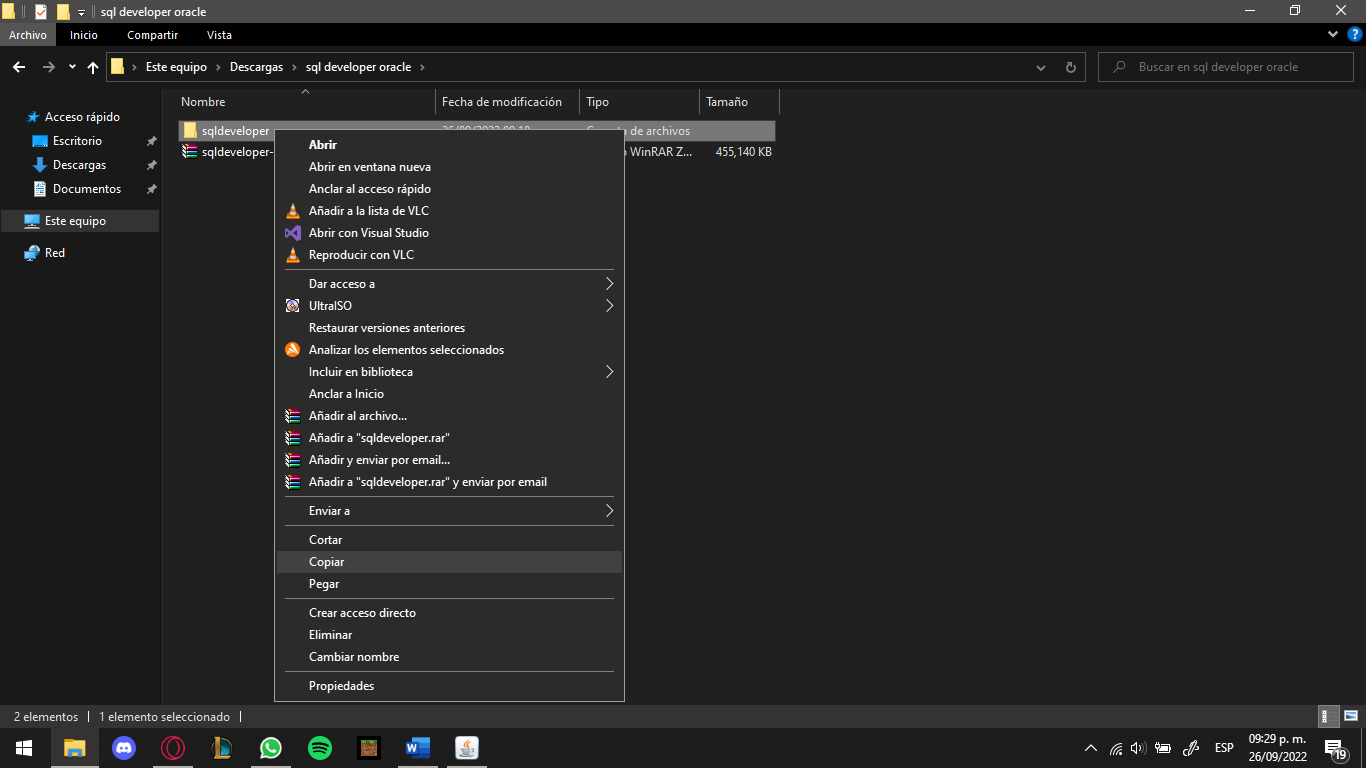
Descripción generada automáticamente

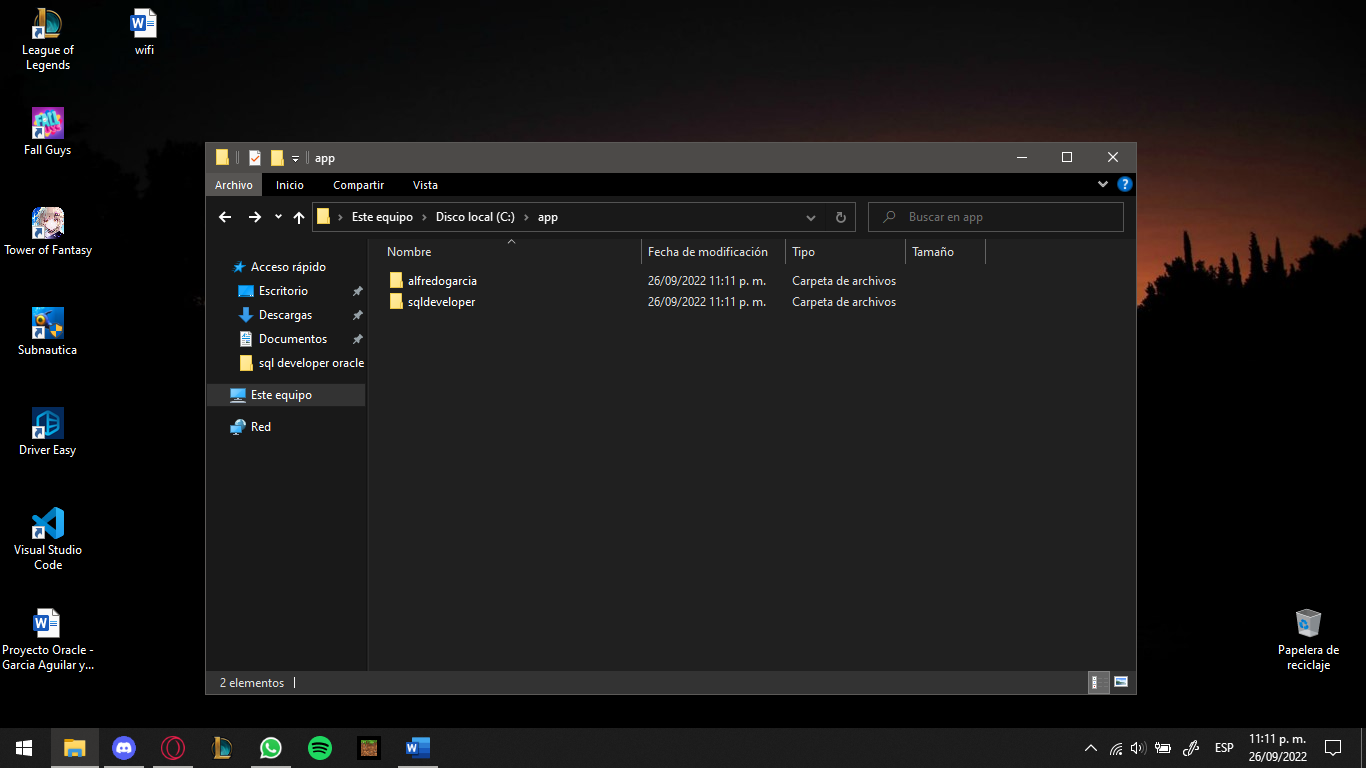
Una vez iniciemos sesión se nos comenzara la descarga, una vez descargado le damos extraer:

Interfaz de usuario gráfica, Texto, Aplicación

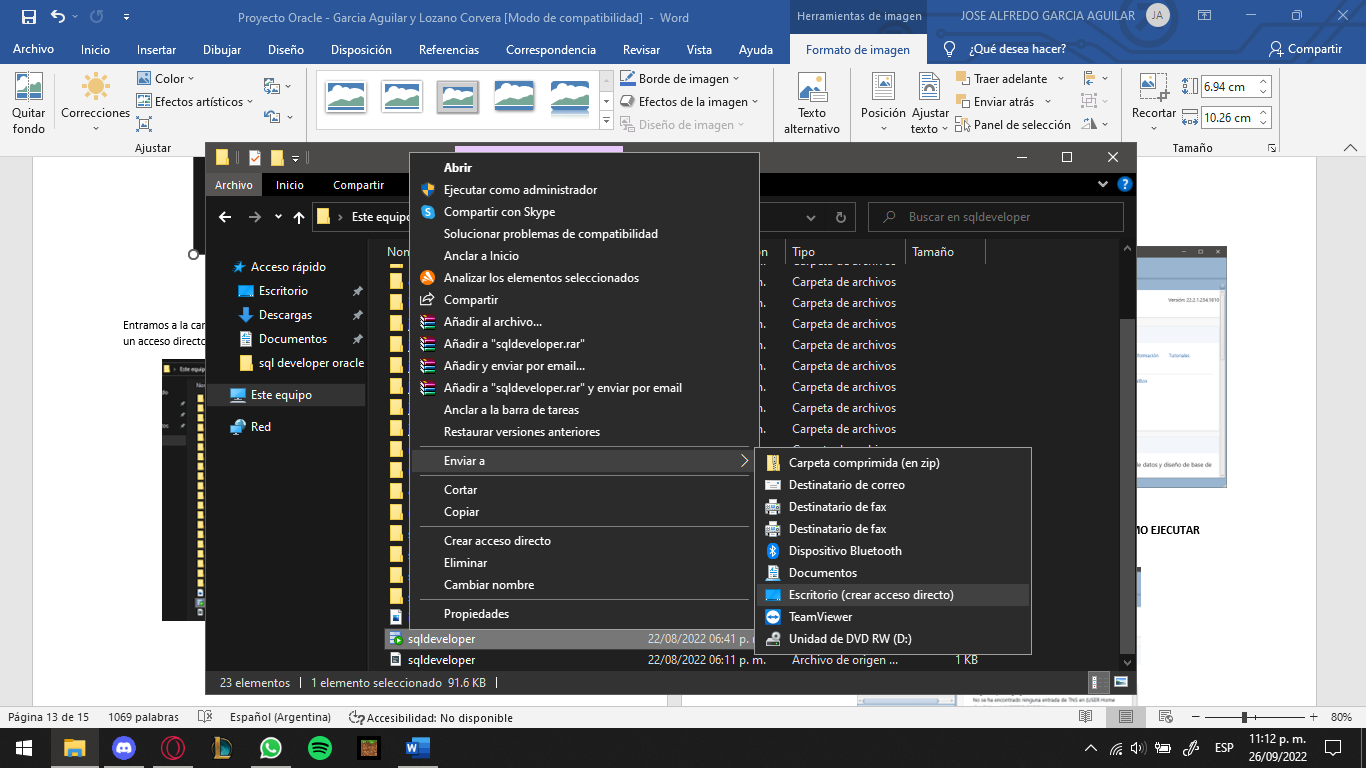
Descripción generada automáticamente

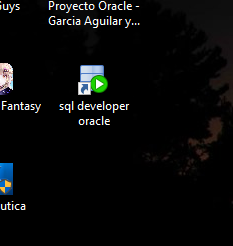
Una vez descomprimido, copiamos el archivo sqldeveloper y lo pegamos en la carpeta de app en nuestro ordenador



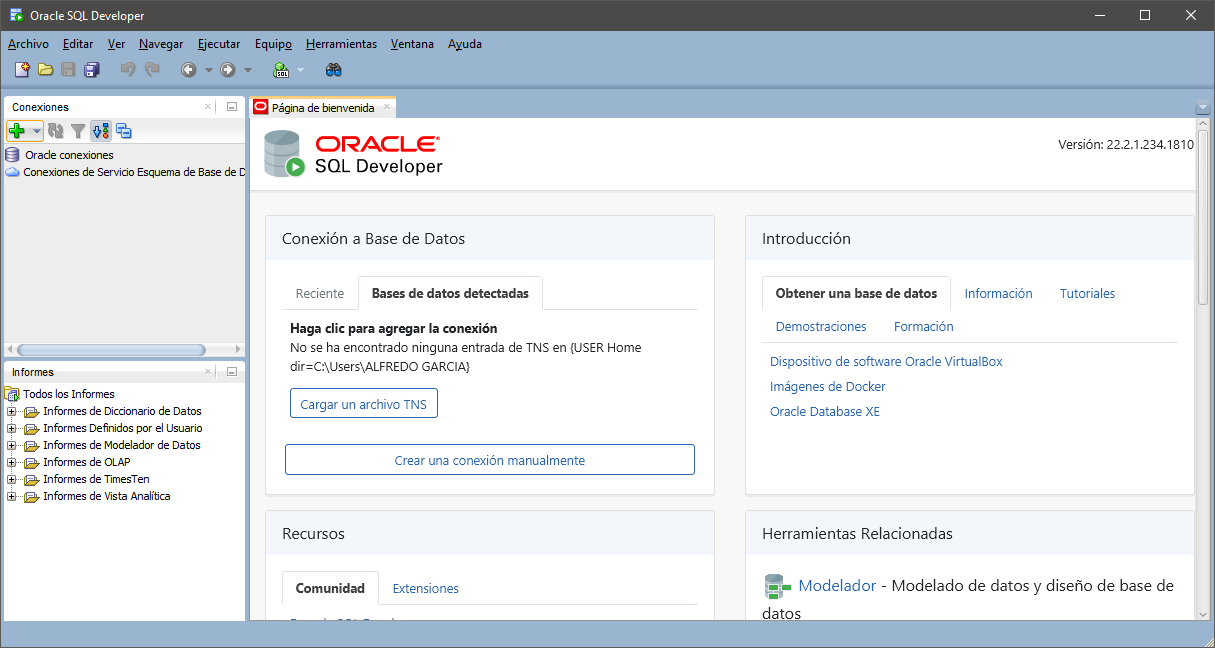


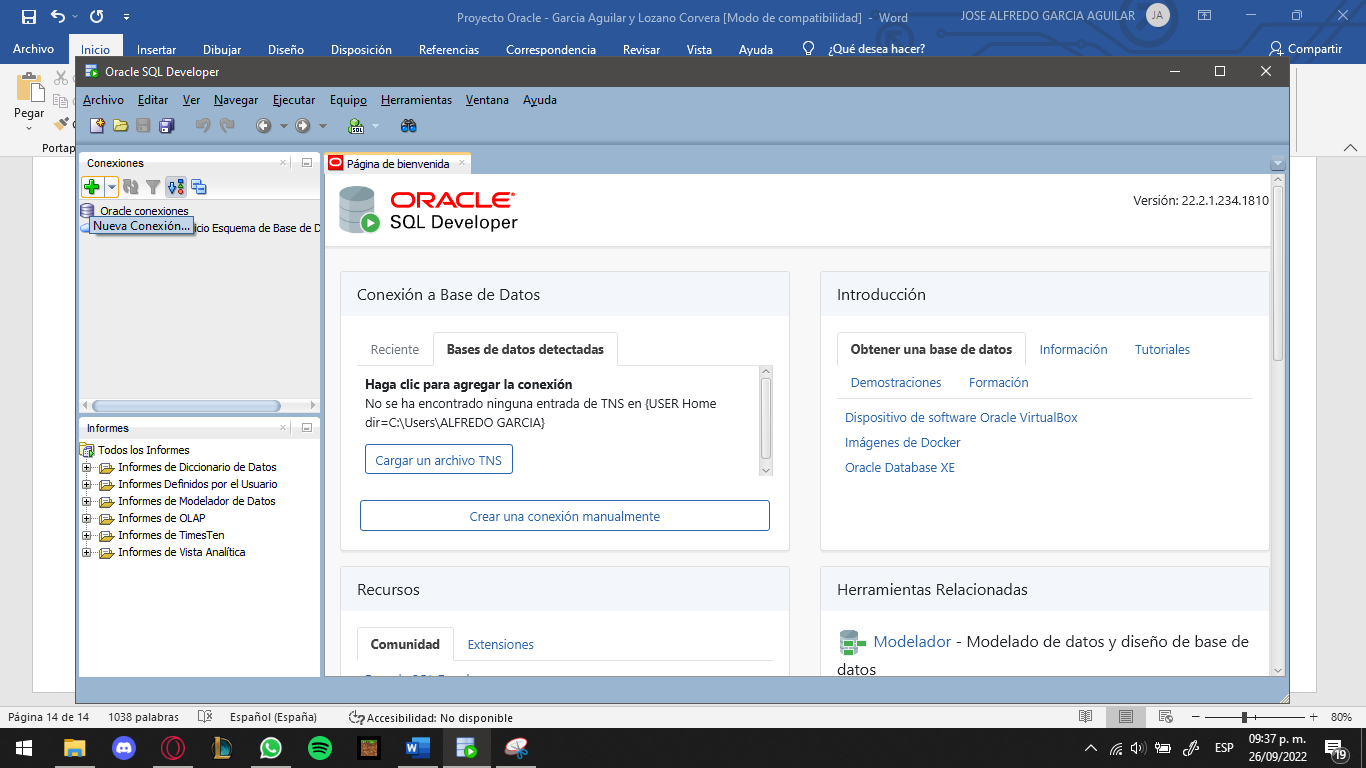
Entramos a la carpeta, hacemos click derecho en el ssqldeveloper.exe ejecutable y ponemos un acceso directo en el escritorio

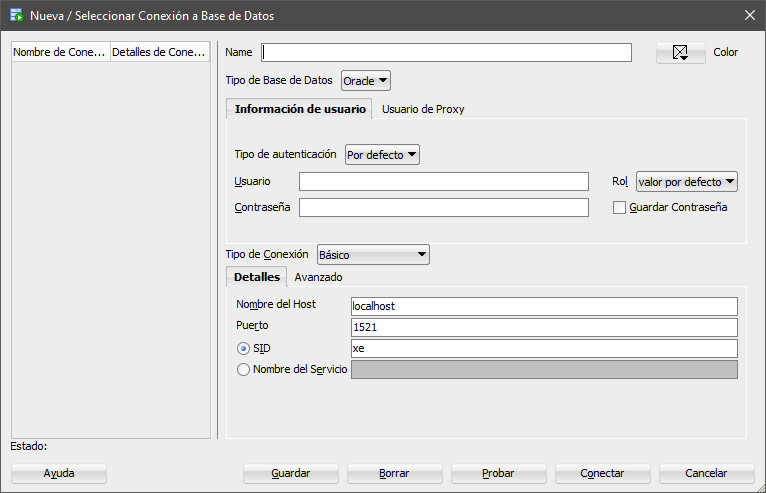




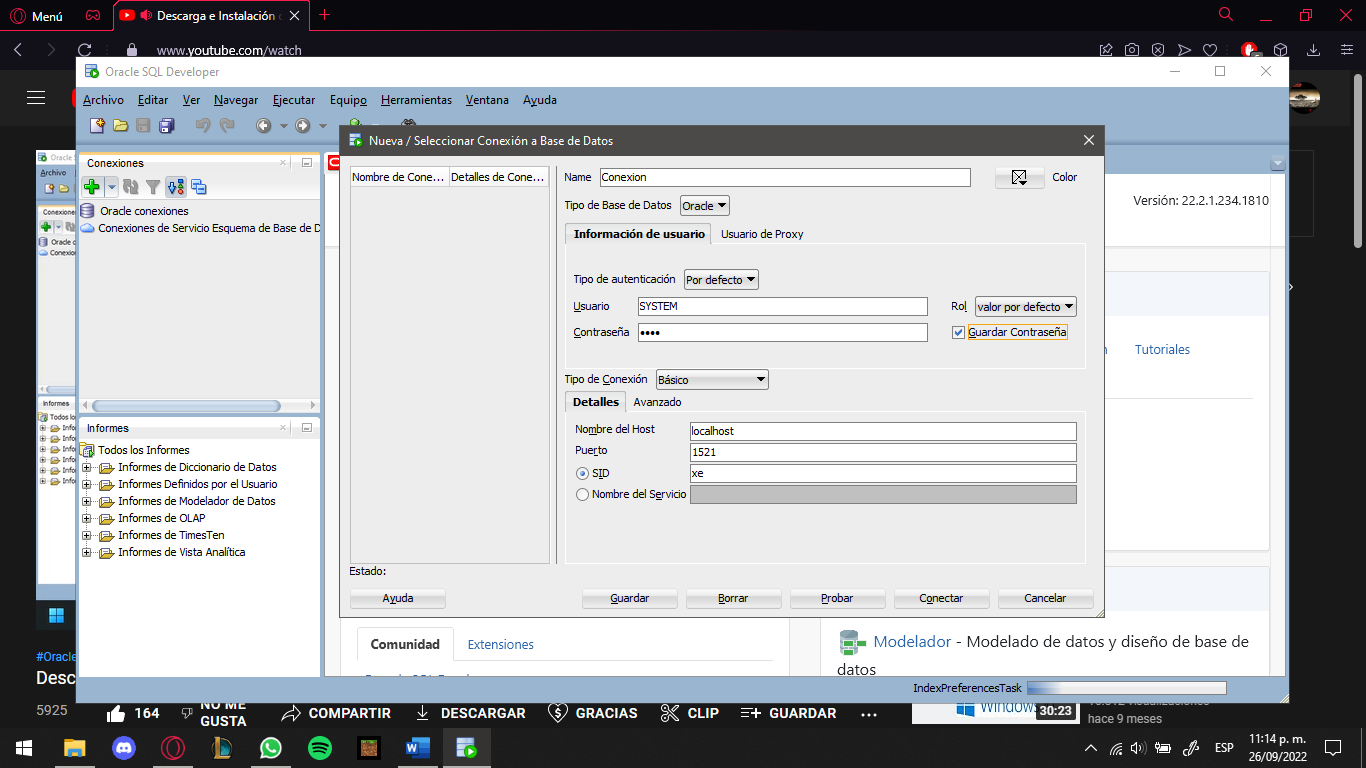
Ya podemos entrar

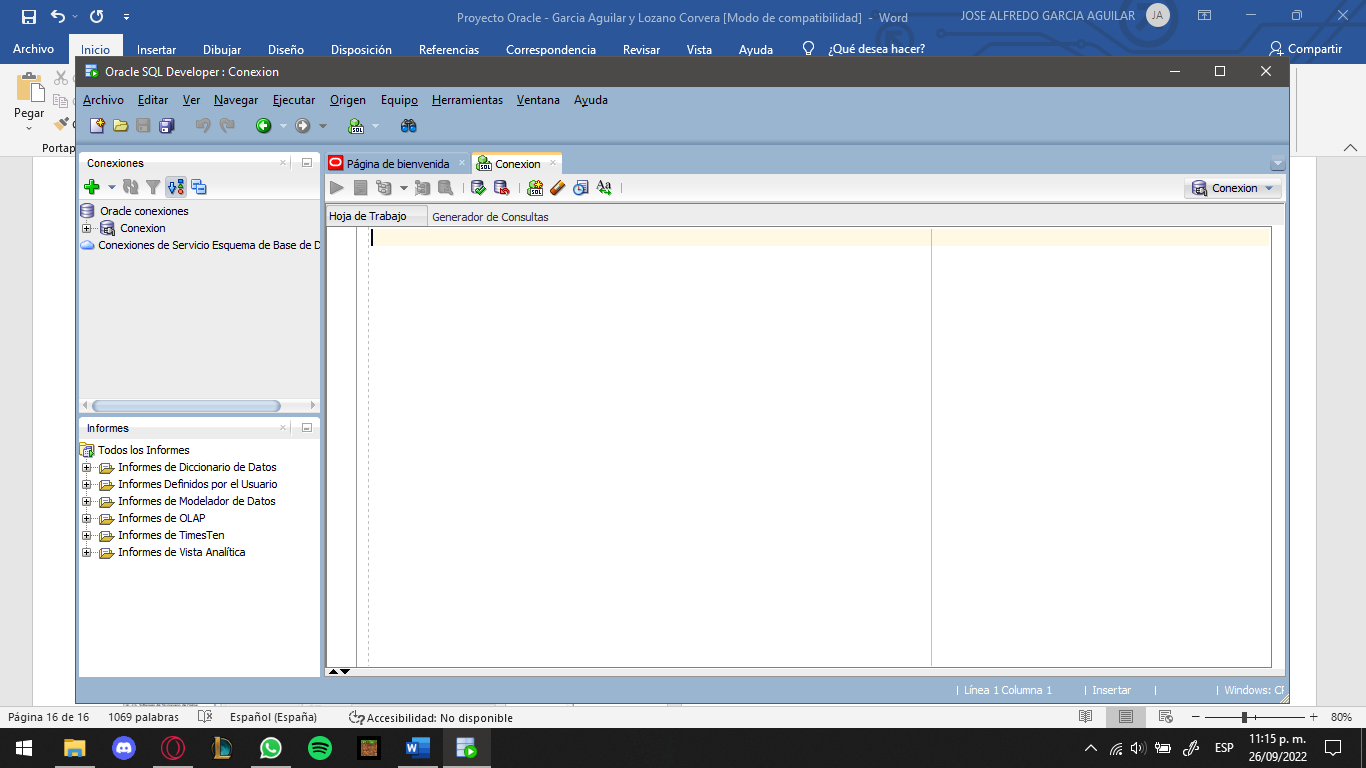


1. **MANUAL DE COMO REALIZAR UNA CONEXIÓN AL SERVIDOR Y COMO EJECUTAR INSTRUCCIONES SQL**



Ingresamos los datos. Ponemos un nombre de conexión, el nombre del usuario predeterminado como SYSTEM, la clave que dimos 1234, guardamos la contraseña en la palomita, damos Probar, Guardar y Conectar



Ya estamos conectados

**CÓMO EJECUTAR INSTRUCCIONES:**

Seleccionamos la instrucción a ejecutar y damos click en la flechita verde

Abajo se comprueba que se ejecutó de forma correcta

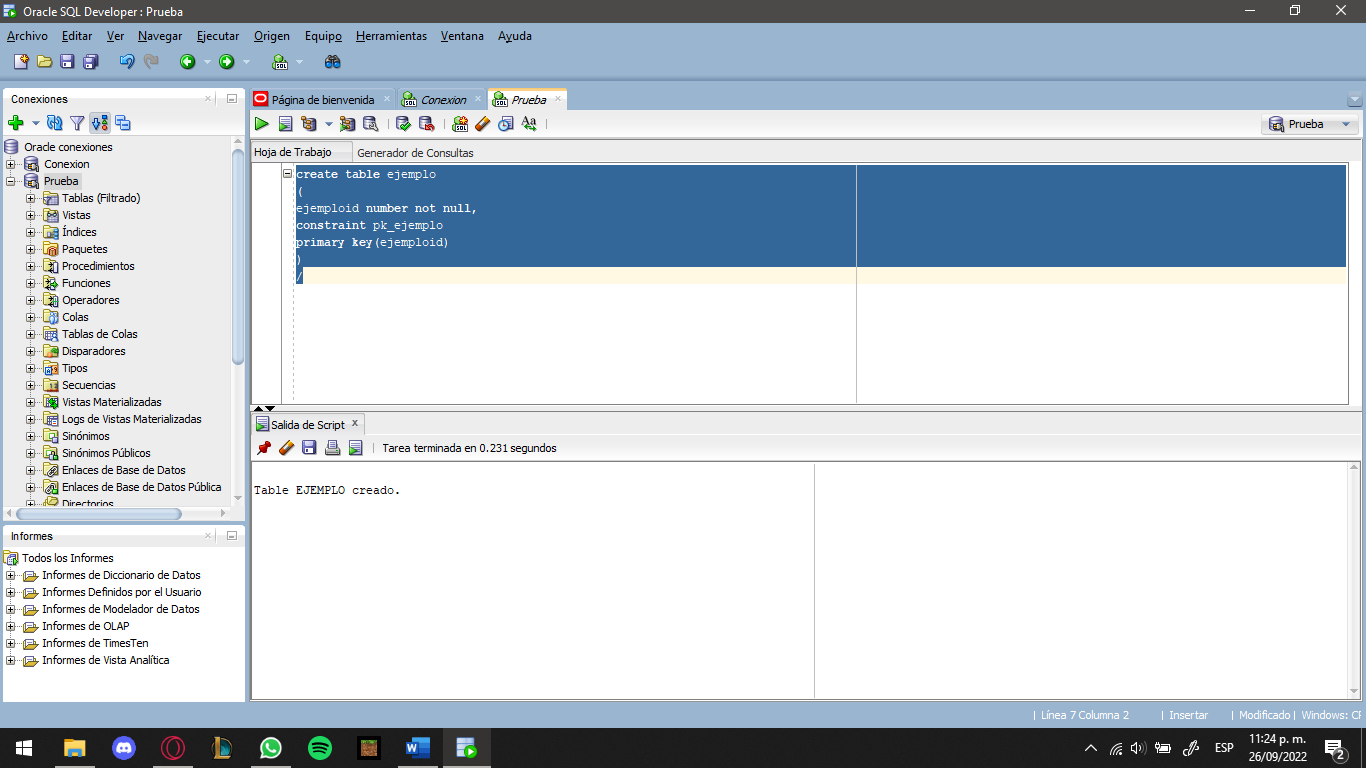
create table ejemplo

(

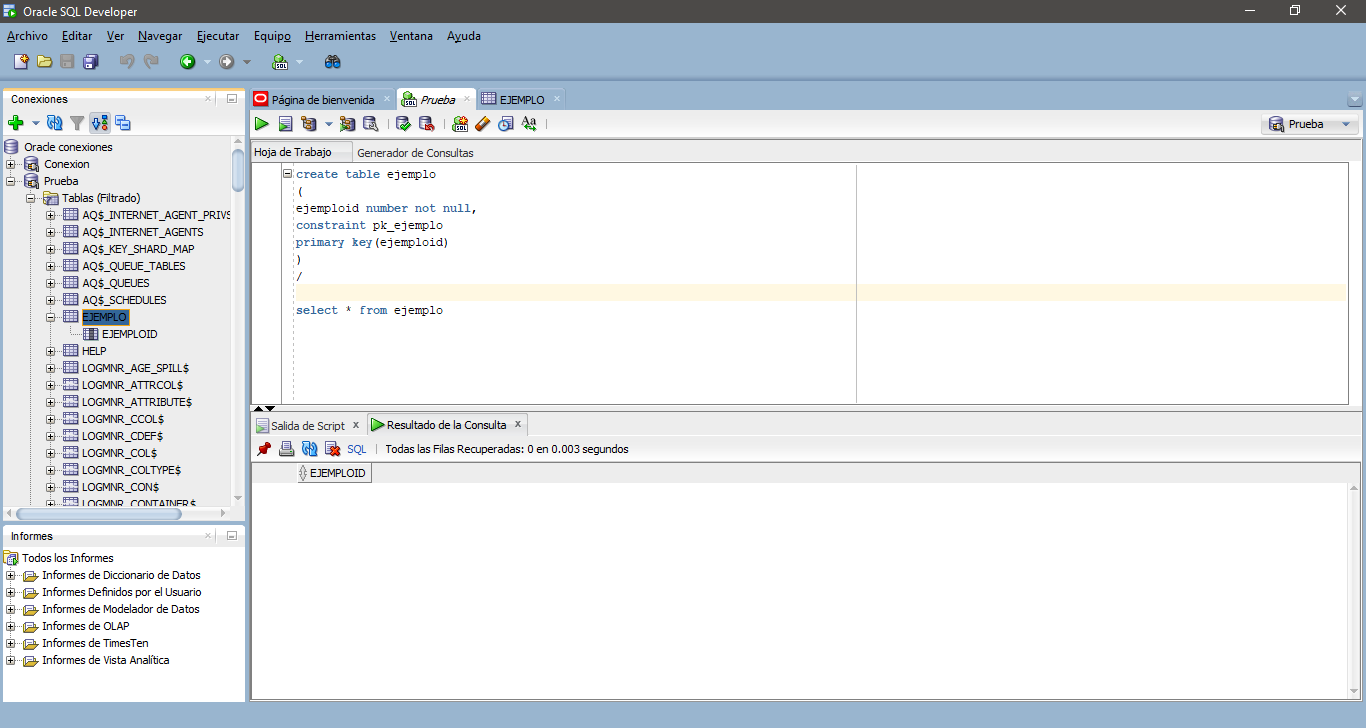
ejemploid number not null,

constraint pk\_ejemplo

primary key(ejemploid)

)

/



1. **CREACIÓN DE USUARIOS Y OTORGARLE PERMISO DE EJECUTAR SOLO SELECT/INSERT**

alter session set "\_ORACLE\_SCRIPT"=true;

CREATE USER usuario1 IDENTIFIED BY clave;

grant select any table to usuario1;

grant insert any table to usuario1;

grant create session to usuario1;

--drop user usuario1;

select \* from system.categories; --permite hacer select

insert into system.categories(categoryid,categoryname,description) values (9,'pescaderia','mojarrita'); --permite hacer insert

insert into system.categories(categoryid,categoryname,description) values (10,'verduras','lechuga'); --permite hacer insert

delete from system.categories where categoryid = 7; --no permite eliminar (privilegios insuficientes)

update system.categories set categoryname = 'salchichoneria' where categoryid = 3; --no permite editar (privilegios insuficientes)

1. **SCRIPT PARA CARGAR LA BASE DE DATOS NORTHWIND (RESUMEN CON LOS 20 TIPOS DE DATOS QUE CAMBIARON)**

CREATE TABLE Categories (

CategoryID INTEGER,

CategoryName NVARCHAR2 (15) NOT NULL,

Description CLOB);

ALTER TABLE Categories

ADD CONSTRAINT PK\_Categories PRIMARY KEY

(

CategoryID

);

CREATE TABLE Suppliers (

SupplierID INTEGER,

CompanyName NVARCHAR2 (40) NOT NULL,

ContactName NVARCHAR2 (30) NULL,

ContactTitle NVARCHAR2 (30) NULL,

Address NVARCHAR2 (60) NULL,

City NVARCHAR2 (15) NULL,

Region NVARCHAR2 (15) NULL,

PostalCode NVARCHAR2 (10) NULL,

Country NVARCHAR2 (15) NULL,

Phone NVARCHAR2 (24) NULL,

Fax NVARCHAR2 (24) NULL,

HomePage CLOB NULL);

ALTER TABLE Suppliers

ADD CONSTRAINT pk\_Suppliers PRIMARY KEY

(

SupplierID

);

CREATE TABLE Products (

ProductID INTEGER,

ProductName NVARCHAR2 (40) NOT NULL,

SupplierID INT NULL,

CategoryID INT NULL,

QuantityPerUnit NVARCHAR2 (20) NULL,

UnitPrice DECIMAL(7,2) NULL,

UnitsInStock SMALLINT NULL,

UnitsOnOrder SMALLINT NULL,

ReorderLevel SMALLINT NULL,

Discontinued NUMBER NOT NULL);

ALTER TABLE Products ADD CONSTRAINT PK\_Products PRIMARY KEY(ProductID);

ALTER TABLE Products ADD CONSTRAINT FK\_Products\_Categories FOREIGN KEY (CategoryID) REFERENCES Categories (CategoryID);

ALTER TABLE Products ADD CONSTRAINT FK\_Products\_Suppliers FOREIGN KEY (SupplierID) REFERENCES Suppliers (SupplierID);

ALTER TABLE Products ADD CONSTRAINT CK\_Products\_UnitPrice CHECK (UnitPrice >= 0);

ALTER TABLE Products ADD CONSTRAINT CK\_ReorderLevel CHECK (ReorderLevel >= 0);

ALTER TABLE Products ADD CONSTRAINT CK\_UnitsInStock CHECK (UnitsInStock >= 0);

ALTER TABLE Products ADD CONSTRAINT CK\_UnitsOnOrder CHECK (UnitsOnOrder >= 0);

CREATE TABLE Employees (

EmployeeID INTEGER,

LastName NVARCHAR2 (20) NOT NULL,

FirstName NVARCHAR2 (10) NOT NULL,

Title NVARCHAR2 (30) NULL,

TitleOfCourtesy NVARCHAR2 (25) NULL,

BirthDate NVARCHAR2 (60) NULL,

HireDate NVARCHAR2 (60) NULL,

Address NVARCHAR2 (60) NULL,

City NVARCHAR2 (15) NULL,

Region NVARCHAR2 (15) NULL,

PostalCode NVARCHAR2 (10) NULL,

Country NVARCHAR2 (15) NULL,

HomePhone NVARCHAR2 (24) NULL,

Extension NVARCHAR2 (4) NULL,

Notes CLOB NULL,

ReportsTo INT NULL,

PhotoPath NVARCHAR2 (255) NULL);

ALTER TABLE Employees ADD CONSTRAINT PK\_Employees PRIMARY KEY (EmployeeID);

ALTER TABLE Employees ADD CONSTRAINT FK\_Employees\_Employees FOREIGN KEY (ReportsTo) REFERENCES Employees (EmployeeID);

CREATE TABLE Customers (

CustomerID NCHAR(5) NOT NULL,

CompanyName NVARCHAR2 (40) NOT NULL,

ContactName NVARCHAR2 (30) NULL,

ContactTitle NVARCHAR2 (30) NULL,

Address NVARCHAR2 (60) NULL,

City NVARCHAR2 (15) NULL,

Region NVARCHAR2 (15) NULL,

PostalCode NVARCHAR2 (10) NULL,

Country NVARCHAR2 (15) NULL,

Phone NVARCHAR2 (24) NULL,

Fax NVARCHAR2 (24) NULL);

ALTER TABLE Customers ADD CONSTRAINT PK\_Customers PRIMARY KEY (CustomerID) ;

CREATE TABLE Shippers (

ShipperID INTEGER,

CompanyName NVARCHAR2 (40) NOT NULL,

Phone NVARCHAR2 (24) NULL);

ALTER TABLE Shippers ADD CONSTRAINT PK\_Shippers PRIMARY KEY (ShipperID);

CREATE TABLE Orders(

OrderID INTEGER,

CustomerID NCHAR (5) NULL,

EmployeeID INT NULL,

OrderDate NVARCHAR2 (60) NULL,

RequiredDate NVARCHAR2 (60) NULL,

ShippedDate NVARCHAR2 (60) NULL,

ShipVia INT NULL,

Freight DECIMAL(7,2) NULL,

ShipName NVARCHAR2 (40) NULL,

ShipAddress NVARCHAR2 (60) NULL,

ShipCity NVARCHAR2 (15) NULL,

ShipRegion NVARCHAR2 (15) NULL,

ShipPostalCode NVARCHAR2 (10) NULL,

ShipCountry NVARCHAR2 (15) NULL);

ALTER TABLE Orders ADD CONSTRAINT PK\_Orders PRIMARY KEY (OrderID);

ALTER TABLE Orders ADD CONSTRAINT FK\_Orders\_Customers FOREIGN KEY (CustomerID) REFERENCES Customers (CustomerID);

ALTER TABLE Orders ADD CONSTRAINT FK\_Orders\_Employees FOREIGN KEY (EmployeeID) REFERENCES Employees (EmployeeID);

ALTER TABLE Orders ADD CONSTRAINT FK\_Orders\_Shippers FOREIGN KEY (ShipVia) REFERENCES Shippers (ShipperID);

CREATE TABLE OrderDetails (

OrderID INT NOT NULL,

ProductID INT NOT NULL,

UnitPrice DECIMAL(7,2) NOT NULL,

Quantity SMALLINT NOT NULL,

Discount REAl NOT NULL);

ALTER TABLE OrderDetails ADD CONSTRAINT PK\_OrderDetails PRIMARY KEY (OrderID, ProductID);

ALTER TABLE OrderDetails ADD CONSTRAINT FK\_Order\_Details\_Orders FOREIGN KEY (OrderID) REFERENCES Orders(OrderID);

ALTER TABLE OrderDetails ADD CONSTRAINT FK\_Order\_Details\_Products FOREIGN KEY (ProductID) REFERENCES Products(ProductID);

ALTER TABLE OrderDetails ADD CONSTRAINT CK\_Discount CHECK (Discount >= 0 and (Discount <= 1));

ALTER TABLE OrderDetails ADD CONSTRAINT CK\_Quantity CHECK (Quantity > 0);

ALTER TABLE OrderDetails ADD CONSTRAINT CK\_UnitPrice CHECK (UnitPrice >= 0);

CREATE TABLE Region(

RegionID INT NOT NULL,

RegionDescription NCHAR (50) NOT NULL

);

CREATE TABLE Territories(

TerritoryID INT,

TerritoryDescription NCHAR (50) NOT NULL,

RegionID INT NOT NULL

);

CREATE TABLE EmployeeTerritories(

EmployeeID INT NOT NULL,

TerritoryID INT

);

ALTER TABLE Region

ADD CONSTRAINT PK\_Region PRIMARY KEY (RegionID);

ALTER TABLE Territories

ADD CONSTRAINT PK\_Territories PRIMARY KEY (TerritoryID);

ALTER TABLE Territories

ADD CONSTRAINT FK\_Territories\_Region FOREIGN KEY (RegionID) REFERENCES Region (RegionID);

ALTER TABLE EmployeeTerritories

ADD CONSTRAINT FK\_EmployeeTerritories PRIMARY KEY (EmployeeID,TerritoryID);

ALTER TABLE EmployeeTerritories

ADD CONSTRAINT FK\_EmployeeTerritories\_Employees FOREIGN KEY (EmployeeID) REFERENCES Employees (EmployeeID);

ALTER TABLE EmployeeTerritories

ADD CONSTRAINT FK\_EmployeeTerritories\_Territories FOREIGN KEY (TerritoryID) REFERENCES Territories (TerritoryID);

-- The following adds constraints to the Northwindm database

CREATE TABLE CustomerCustomerDemo(

CustomerID NCHAR (5) NOT NULL,

CustomerTypeID NCHAR (10) NOT NULL

);

CREATE TABLE CustomerDemographics(

CustomerTypeID NCHAR (10) NOT NULL ,

CustomerDesc CLOB NULL

);

ALTER TABLE CustomerCustomerDemo

ADD CONSTRAINT PK\_CustomerCustomerDemo PRIMARY KEY (CustomerID,CustomerTypeID);

ALTER TABLE CustomerDemographics

ADD CONSTRAINT PK\_CustomerDemographics PRIMARY KEY (CustomerTypeID);

ALTER TABLE CustomerCustomerDemo

ADD CONSTRAINT FK\_CustomerCustomerDemo FOREIGN KEY (CustomerTypeID) REFERENCES CustomerDemographics (CustomerTypeID);

ALTER TABLE CustomerCustomerDemo

ADD CONSTRAINT FK\_CustomerCustomerDemo\_Customers FOREIGN KEY(CustomerID) REFERENCES Customers (CustomerID);

1. **SCRIPT PARA CREAR LA FAMILIA DE VISTAS DE LA BASE DE DATOS NORTHWIND**

create view vw\_products as

select p.productid, p.productname, p.quantityperunit, p.unitprice as produnitprice,

p.unitsinstock, p.unitsonorder, p.reorderlevel, p.discontinued,

s.supplierid, s.companyname, s.contactname, s.contacttitle, s.address, s.city, s.region,

s.postalcode, s.country, s.phone, s.fax, s.homepage, c.categoryid, c.categoryname, c.description

from products p

inner join suppliers s on p.supplierid = s.supplierid

inner join categories c on p.categoryid = c.categoryid

;

create view vw\_orders as

select o.orderid, o.orderdate, o.requireddate, o.shippeddate, o.freight, o.shipname, o.shipaddress, o.shipcity,

o.shipregion, s.shipperid, s.companyname as nomcomenvio, s.phone as envphone, c.customerid, c.companyname as nomcliente,

c.contactname as ctecontactname, c.contacttitle as ctecontacttitle,

c.address as cteaddress, c.city as ctecity, c.region as cteregion, c.postalcode as ctepostalcode,

c.country as ctecountry, c.phone as ctephone, c.fax as ctefax, e.employeeid, e.lastname, e.firstname,

e.title, e.titleofcourtesy, e.birthdate, e.hiredate,

e.address as empaddress, e.city as empcity, e.region as empregion, e.postalcode as emppostalcode,

e.country as empcountry, e.homephone, e.extension, e.reportsto

from orders o

inner join shippers s on o.shipvia = s.shipperid

inner join customers c on o.customerid = c.customerid

inner join employees e on o.employeeid = e.employeeid

;

create view vw\_order\_details as

select d.quantity, d.unitprice, d.discount, p.\*, o.\*

from orderdetails d

inner join vw\_products p on d.productid = p.productid

inner join vw\_orders o on d.orderid = o.orderid

;

--select \* from vw\_products;

--select \* from vw\_orders;

--select \* from vw\_order\_details;

--select count(distinct productid) from vw\_products

--select count(distinct orderid) from vw\_orders

--select count(distinct productid) from vw\_order\_details

--drop view vw\_products;

--drop view vw\_orders;

--drop view vw\_order\_details;

1. **SCRIPT CON EL PROCEDIMIENTO ALMACENADO QUE INSERTE Y MODIFIQUE LA TABLA TERRITORIES**

create or replace procedure sp\_mttoterritories (territoryid\_v out varchar2, territorydescription\_v in char, regionid\_v in int)

as

begin

declare

contador varchar(40);

begin

select count(\*) into contador from territories where territoryid = territoryid\_v;

if (contador > 0 ) then

update territories set territoryid = territoryid\_v, territorydescription = territorydescription\_v, regionid = regionid\_v

where territoryid = territoryid\_v;

else

select max(territoryid) into territoryid\_v from territories;

insert into territories(territoryid, territorydescription, regionid)

values (territoryid\_v, territorydescription\_v, regionid\_v);

end if;

exception when no\_data\_found then

dbms\_output.put\_line('error al actualizar la tabla territories');

end;

end;

/

--drop procedure sp\_mttoterritories;

--select \* from territories;

begin

sp\_mttoterritories(113, 'ESCUELA', 1);

commit;

end;

--select \* from territories where territoryid = '113';

1. **SCRIPT CON UN TRIGGER QUE NO PERMITA ELIMINAR TERRITORIES**

create or replace trigger tr\_eliminar\_territories

before delete on territories

for each row

begin

raise\_application\_error(-20001,'Por el momento no se puede eliminar registros');

end tr\_eliminar\_territories;

--select \* from territories;

--delete from territories where TerritoryID = '2903';

--select \* from territories where territoryid = '2903';

--drop trigger tr\_eliminar\_territories;

**NOTA: LA UNICA MANERA DE BORRAR UNA BASE DE DATOS USANDO ORACLE ES DESINSTALANDO E INSTALANDO EL PROGRAMA, ASÍ QUE USAREMOS EL SIGUIENTE SCRIPT PARA ESTAR ELIMINANDO LAS TABLAS CADA VEZ QUE NECESITAMOS REUSAR LA NORTHWIND:**

drop table orderdetails;

drop table employeeterritories;

drop table customercustomerdemo;

drop table customerdemographics;

drop table products;

drop table orders;

drop table territories;

drop table categories;

drop table suppliers;

drop table employees;

drop table customers;

drop table shippers;

drop table region;