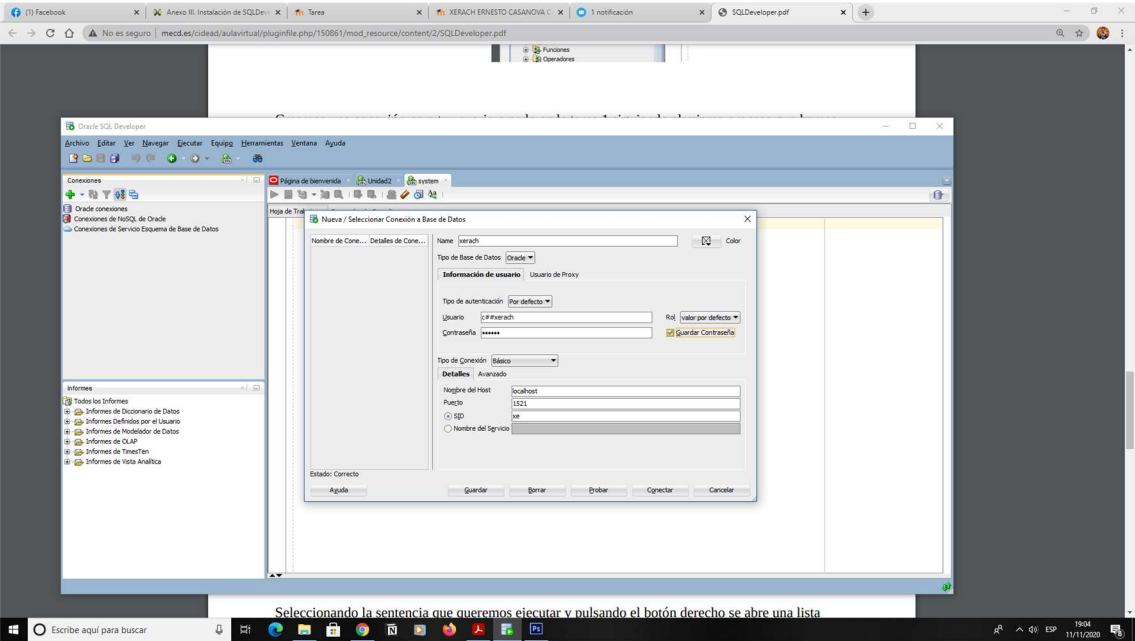


MODIFICAR: NOMBRE DE LAS RESTRICCIONES DE LA TABLA PRODUCTO EN EL PRIMER EJERCICIO Y CAPTURAS SEGUNDO EJERCICIO (Y UN VARCHAR2)

Realiza los ejercicios en la Hoja de trabajo de SQLDeveloper o desde SQLPlus conectando con tu usuario, creado en la primera tarea, y comprueba que las sentencias son correctas antes de incluirlas en el script. Una vez hayas hecho todas para probar el script, desde SQLPlus, ejecuta start nombrescript.sql o @nombrescript.sql eliminando previamente todas las tablas creadas para poder crearlas de nuevo.

Abro el SQL Developer y conecto con el usuario creado en la Unidad 1.



Ejercicio 1.

Vamos a crear las tablas para una tienda virtual que distribuye productos agrupados en familias en varias tiendas.

Realiza un script llamado Creatienda.sql que implemente los ejercicios descritos a continuación.

Precede cada una de las sentencias SQL de los ejercicios con un comentario que incluya el enunciado del ejercicio correspondiente. Recuerda que los comentarios van precedidos del símbolo -- al inicio de la línea.

Con las sentencias DDL de SQL crea la tablas especificadas a continuación aplicando las restricciones (constraints) pedidas. Se debe cumplir la integridad referencial.

TABLA FAMILIA: Contiene las familias a las que pertenecen los productos, como por ejemplo ordenadores, impresoras,etc.

Nombre Columna	Descripción	Tipo dato	Restricciones
Codfamilia	Código que distingue una familia de otra	Numérico de 3 dígitos	Clave primaria.
Denofamilia	Denominación de la familia	Alfanumérico de 50 caracteres	No puede haber dos familias con la misma denominación.

			Debe tener contenido.
--	--	--	-----------------------

Creo la tabla FAMILIA con dos columnas (**Codfamilia y Denofamilia**):

- Indico que el tipo de dato para **Codfamilia** es numérico de tres dígitos **NUMBER(3)** y además indico que será la clave primaria **PRIMARY KEY** mediante una restricción **CONSTRAINT** a la que le he dado nombre de **Familia_PK**.
- El tipo de dato para **Denofamilia** es un alfanumérico de 50 caracteres y se indica con **VARCHAR(50)**. Además, al no poder haber dos familias con la misma denominación indico que este será un campo **UNIQUE** mediante una restricción **CONSTRAINT** a la que he dado el nombre de **Familia_UK**. Por último indico que este campo siempre debe tener contenido con **NOT NULL**.
- Para facilitar la gestión posterior de la estructura de la base de datos, he añadido comentarios tanto a las tablas como a las columnas con la instrucción **COMMENT ON TABLE/COLUMN**, con la sintaxis que se muestra en la captura. Todo lo que está entre comillas simple es el comentario.

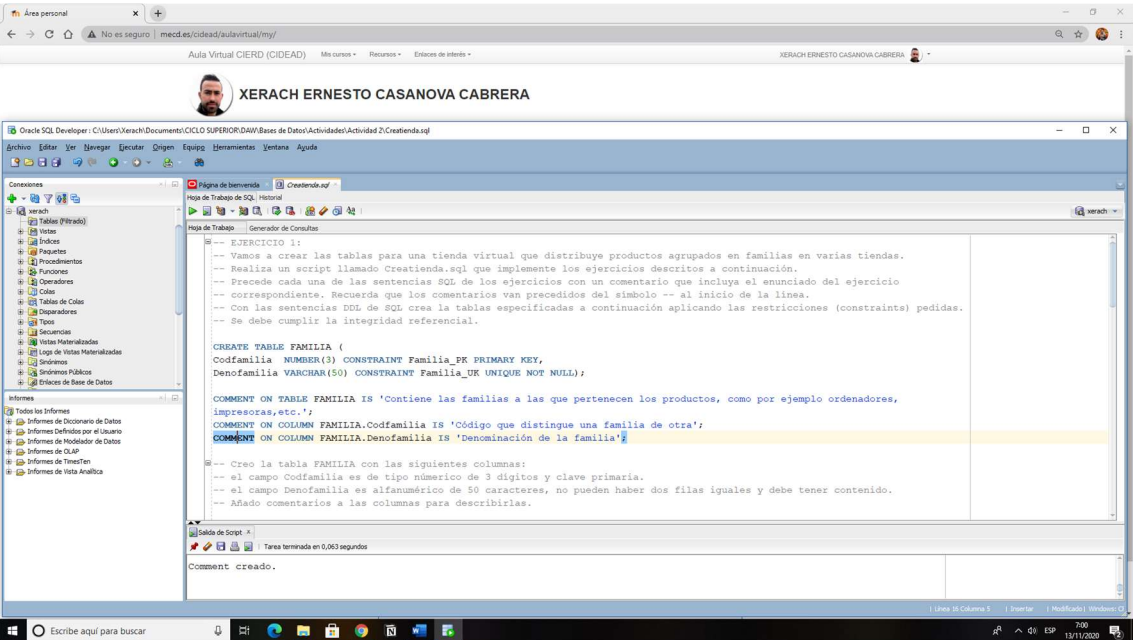


TABLA PRODUCTO => contendrá información general sobre los productos que distribuye la empresa a las tiendas.

Nombre Columna	Descripción	Tipo dato	Restricciones
Codproducto	Código que distingue un producto de otro	Numérico de 5 dígitos	Clave primaria.
Denoproducto	Denominación del producto	Alfanumérico de 20 caracteres	Debe tener contenido.

Descripcion	Descripción del producto	Alfanumérico de 100 caracteres	
PrecioBase	Precio base del producto	Númérico de 8 dígitos dos de ellos decimales	Mayor que 0. Debe tener contenido.
PorcReposición	Porcentaje de reposición aplicado a ese producto. Se utilizará para aplicar a las unidades mínimas y obtener el número total de unidades a reponer cuando el stock esté bajo mínimo	Númérico de 3 dígitos	Mayor que 0
UnidadesMinimas	Unidades mínimas recomendables en almacén	Númérico de 4 dígitos	Mayor que 0. Debe tener contenido.
Codfamilia	Código de la familia a la que pertenece el producto	Númérico de 3 dígitos	Clave ajena, referencia a Codfamilia de la tabla FAMILIA. Debe tener contenido.

Creo la tabla PRODUCTO con siete columnas (**Codproducto**, **Denoproducto**, **Descripcion**, **PreciobaBase**, **PorcReposicion**, **UnidadesMinimas**, **Codfamilia**):

- **Codfamilia** será un tipo numérico de 5 dígitos **NUMBER(5)**. Este campo también será clave primaria **PRIMARY KEY** y ha sido indicado dándole un nombre a la restricción **CONSTRAINT Producto_PK**.
- **Denoproducto** será una columna de tipo alfanumérico de 20 caracteres **VARCHAR(20)** y su única restricción es que debe tener contenido **NOT NULL**.
- **Descripcion** es una columna de tipo alfanumérico de 100 caracteres **VARCHAR(100)** y no tiene ningún tipo de restricción.
- La columna **PrecioBase** es un tipo de dato numérico de 8 dígitos de los cuales 2 son decimales **NUMBER(8,2)**. Le he añadido restricción **CHECK** para que los datos introducidos sean mayor que 0 **PrecioBase > 0** y a dicha restricción **CONSTRAINT** la he llamado **PreBa_CK**. Por último he añadido la restricción **NOT NULL** para que la columna siempre tenga contenido.
- **PorcReposicion** es un tipo de dato numérico de tres dígitos **NUMBER(3)**. Al igual que la columna PrecioBase, esta tiene una restricción **CHECK (PorcReposicion > 0)** para que solo permita valores mayores que 0. En esta ocasión la restricción se llama PorcRepo_CK.
- **UnidadesMinimas** es un tipo de dato numérico de 4 dígitos **NUMBER(4)** y también tiene la misma restricción **CHECK** de las dos columnas anteriores. Además tampoco se permitirán valores nulos con **NOT NULL**.
- **CodFamilia** es un tipo numérico de 3 dígitos **NUMBER(3)** coincidiendo con la columna CodFamilia de la tabla FAMILIA. Es necesario que coincidan para poder hacer referencia a dicha columna mediante la restricción **FOREIGN KEY**, a la cual he llamado **Prod_Fam_FK** (este

nombre de restricción hace referencia a la tabla PRODUCTO (**Prod**) y a la tabla FAMILIA (**Fam**), indicando que es una clave ajena (**FK**). Por último añado la restricción **NOT NULL** para que siempre tenga contenido.

- Después de la instrucción para crear la tabla, añado un **COMMENT ON** a la tabla y a cada columna.

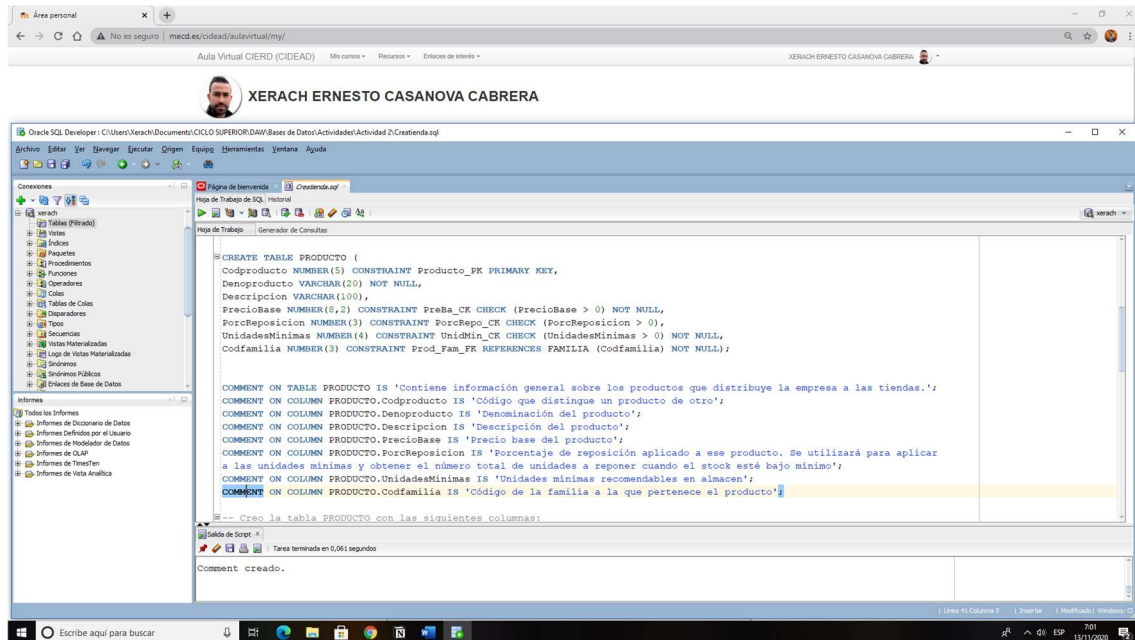


TABLA TIENDA=> contendrá información básica sobre las tiendas que distribuyen los productos.

Nombre Columna	Descripción	Tipo dato	Restricciones
Codtienda	Código que distingue una tienda de otra.	Número de 3 dígitos	Clave primaria.
Denotienda	Denominación o nombre de la tienda.	Alfanumérico de 20 caracteres	Debe tener contenido.
Telefono	Teléfono de la tienda	Alfanumérico de 11 caracteres	
CodigoPostal	Código Postal donde se ubica la tienda	Alfanumérico de 5 caracteres	Debe tener contenido.
Provincia	Provincia donde se ubica la tienda	Alfanumérico de 5 caracteres	Debe tener contenido.

Creo la tabla TIENDA con cinco columnas (**Codtienda**, **Denotienda**, **Telefono**, **CodigoPostal**, **Provincia**):

- **Codtienda** es una columna numérica de 3 dígitos **NUMBER(3)** y además la he denominado como clave primaria de la tabla **PRIMARY KEY** nombrando la restricción **CONSTRAINT** como **Tienda_PK**.
- La columna **Denotienda** será un alfanumérico de 20 caracteres **VARCHAR(20)** y es **NOT NULL**, pues debe tener siempre contenido.
- **Telefono** es una columna de tipo alfanumérico de 11 caracteres **VARCHAR(11)** y no tiene restricciones.
- **CodigoPostal** y Provincia son columnas de tipo alfanumérico de 5 caracteres **VARCHAR(5)** y tiene la restricción **NOT NULL** para que siempre tengan contenido.
- Para terminar, una vez creada la tabla he añadido comentarios a la tabla y sus columnas con la instrucción **COMMENT ON**.

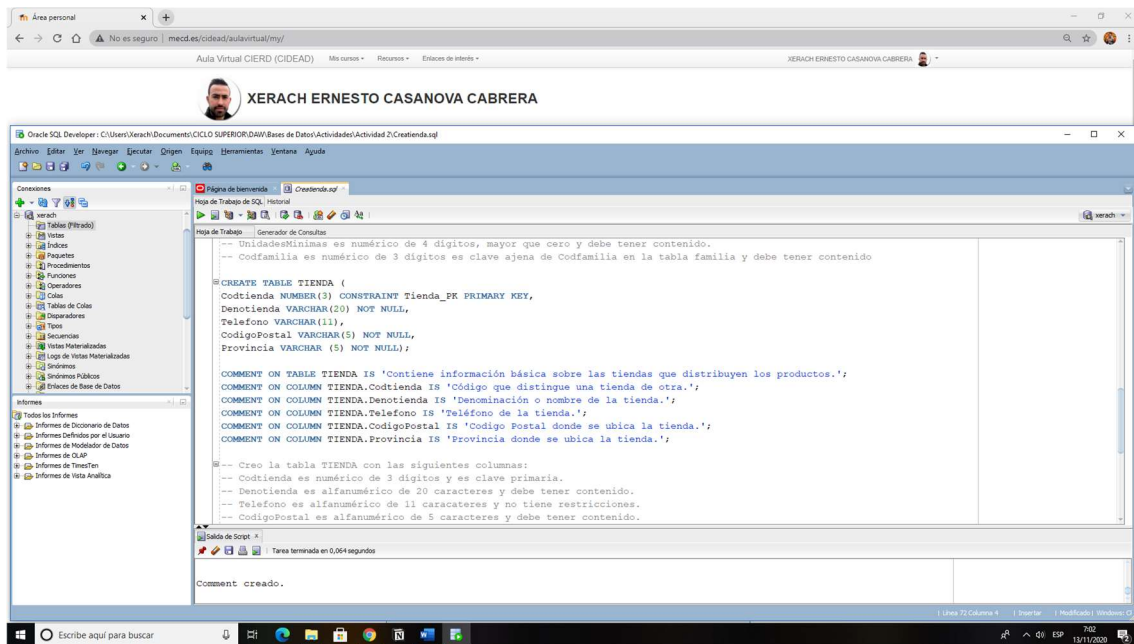


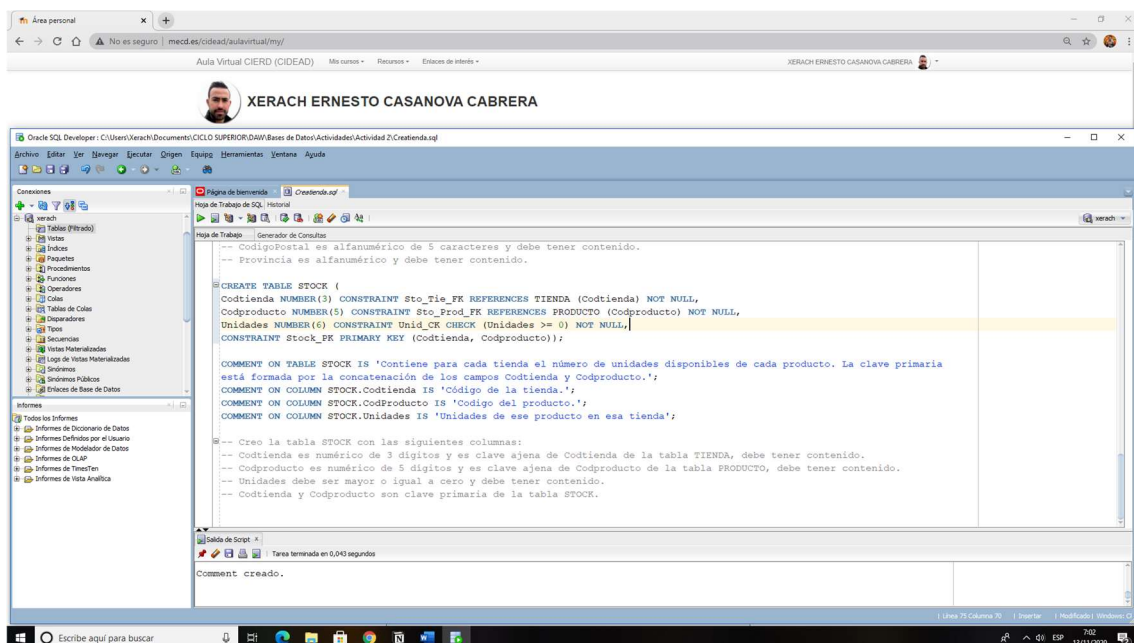
TABLA STOCK => Contendrá para cada tienda el número de unidades disponibles de cada producto. La clave primaria está formada por la concatenación de los campos Codtienda y Codproducto.

Nombre Columna	Descripción	Tipo dato	Restricciones	
Codtienda	Código de la tienda.	Numérico de 3 dígitos	Clave primaria: (Codtienda,Codproducto)	Clave ajena, referencia a Codtienda de la tabla tienda. Debe tener contenido.
Codproducto	Código del producto	Numérico de 5 dígitos	Permite que un producto pueda aparecer en varias tiendas, y que en una tienda puedan haber varios productos.	Clave ajena, referencia a Codproducto

			de la tabla PRODUCTO. Debe tener contenido.
Unidades	Unidades de ese producto en esa tienda	Númerico de 6 dígitos.	Mayor o igual a 0. Debe tener contenido.

Creo la tabla STOCK con tres columnas (**Codtienda**, **Codproducto** y **Unidades**):

- La columna **Codtienda** es tipo numérico de 3 dígitos **NUMBER(3)** y tiene la restricción de clave ajena **FOREIGN KEY** haciendo referencia a la columna Codtienda de la tabla TIENDA. La he llamado **Sto_Tie_FK** (Stock – Tienda – Foreign Key). Por último le he añadido la restricción NOT NULL para que siempre tenga contenido.
- La columna **Codproducto** es tipo numérico de 5 dígitos **NUMBER(5)** y tiene la restricción de clave ajena **FOREIGN KEY** haciendo referencia a la columna Codproducto de la tabla PRODUCTO. La he llamado **Sto_Pro_FK** (Stock – Producto – Foreign Key). Por último le he añadido la restricción NOT NULL para que siempre tenga contenido.
- Las dos columnas anteriores son clave primaria de la propia tabla y eso lo he indicado al final de la instrucción, añadiendo la restricción mediante el comando **CONSTRAINT**. El nombre de la restricción es **Stock_PK**.
- **Unidades** es una columna de tipo numérico de 6 dígitos. Tiene una restricción para que solo se puedan añadir dígitos mayores o igual a cero, lo he hecho con **CHECK (Unidades >= 0)**. Esta restricción la he llamado **Unid_CK**. Por último le añado la restricción **NOT NULL** para que siempre tenga contenido.
- Como en las tablas anteriores, añado al finalizar la creación de la tabla, comentarios para la tabla y cada una de sus columnas con el comando **COMMENT ON**.



EJERCICIO 2:

A) Modificar las tablas creadas en el ejercicio anterior siguiendo las indicaciones. Los ejercicios se incluirán en un script llamado ModificaTienda.sql. Cada uno de ellos, como en el ejercicio anterior, irá precedido de un comentario con el enunciado.

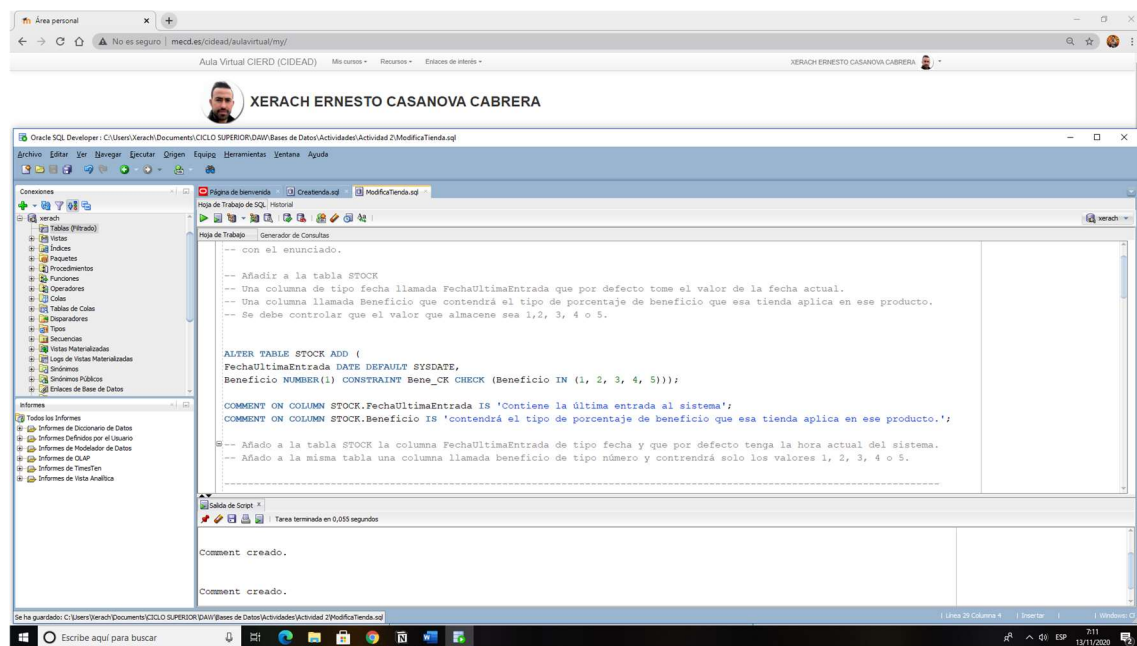
Añadir a la tabla STOCK

Una columna de tipo fecha llamada FechaUltimaEntrada que por defecto tome el valor de la fecha actual.

Una columna llamada Beneficio que contendrá el tipo de porcentaje de beneficio que esa tienda aplica en ese producto. Se debe controlar que el valor que almacene sea 1,2, 3, 4 o 5.

Utilizo **ALTER TABLE STOCK ADD** para añadir columnas a la tabla STOCK:

- **FechaUltimaEntrada** será de tipo fecha **DATE** y le indico la restricción **DEFAULT** para que por defecto siempre tenga la fecha del sistema **SYSDATE**.
- **Beneficio** es una columna de tipo numérico de un solo dígito. Tiene una restricción que solo permitirá valores entre 1 y 5. Para ello he usado **CHECK (Beneficio IN(1, 2, 3, 4, 5))**. A esta restricción la he llamado **Bene_CK**.
- Por último añado un comentario a las nuevas columnas mediante la instrucción **COMMENT ON COLUMN**.



En la tabla PRODUCTO

Eliminar de la tabla producto la columna Descripción.

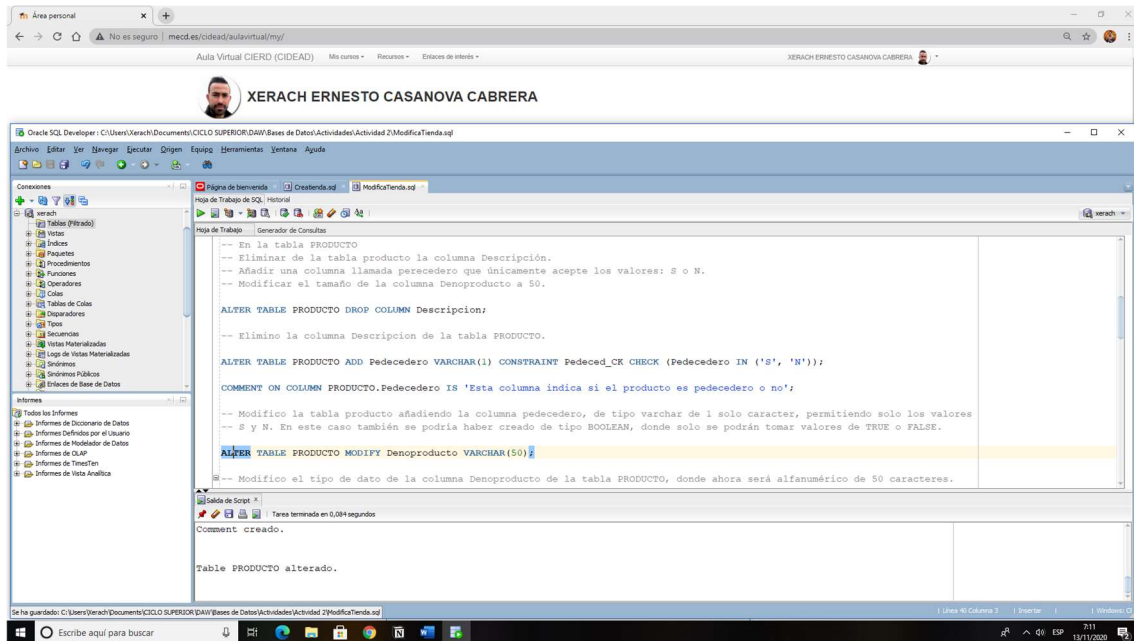
Añadir una columna llamada perecedero que únicamente acepte los valores: S o N.

Modificar el tamaño de la columna Denoproducto a 50.

- Utilizo la instrucción **ALTER TABLE PRODUCTO DROP COLUMN** para eliminar la columna Descripción de la tabla PRODUCTO.
- Añado la columna **Perecedero** a la tabla PRODUCTO con la instrucción **ALTER TABLE PRODUCTO ADD**. La columna se llama Perecedero y será un tipo de dato alfanumérico de un

carácter VARCHAR(1). La restricción **CHECK (Pedecedero IN ('S', 'N'))** permitirá que solo se pueda añadir S o N como dato. A esta restricción la he llamado **PEDECED_CK**. Una vez creado genero un comentario sobre la nueva columna mediante **COMMENT ON COLUMN**

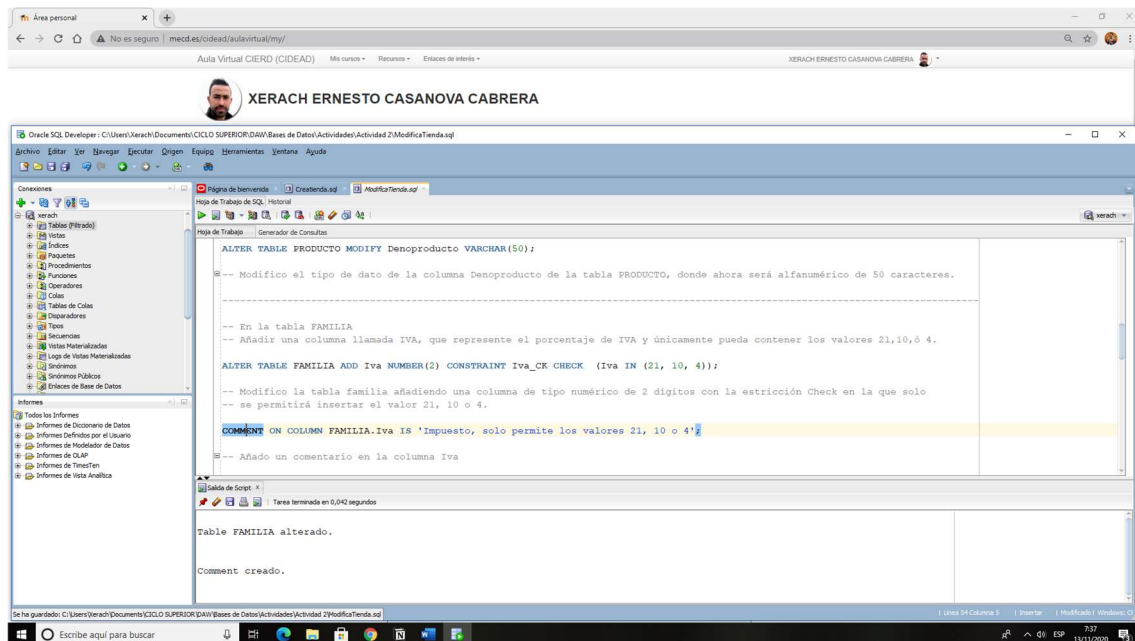
- Utilizo la instrucción **ALTER TABLE PRODUCTO MODIFY** para modificar la columna Denoproducto, permitiendo que pase de tener un máximo de 20 caracteres a un máximo de 50 **VARCHAR(50)**



En la tabla FAMILIA

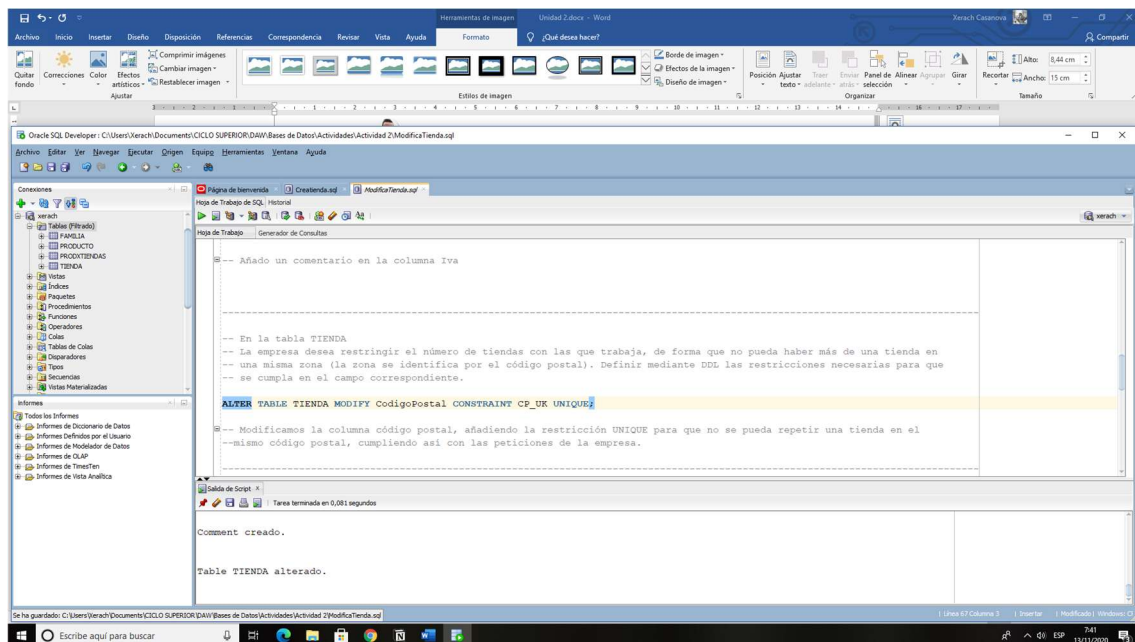
Añadir una columna llamada IVA, que represente el porcentaje de IVA y únicamente pueda contener los valores 21,10,ó 4.

- Con la instrucción SQL **ALTER TABLE FAMILIA ADD** añado una columna llamada **Iva** que será de tipo numérico de dos dígitos **NUMBER(2)** y tendrá una restricción llamada **Iva_CK**, la cual dejará insertar solamente valores que sean 21, 10 o 4 con **CHECK (Iva IN (21,10,4))**;



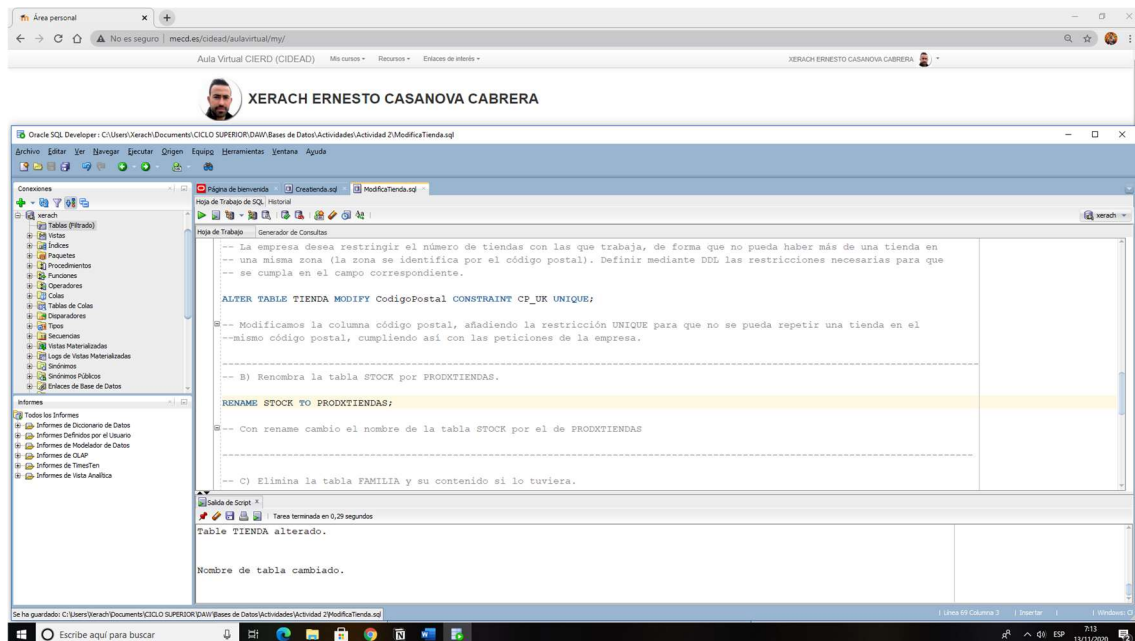
La empresa desea restringir el número de tiendas con las que trabaja, de forma que no pueda haber más de una tienda en una misma zona (la zona se identifica por el código postal). Definir mediante DDL las restricciones necesarias para que se cumpla en el campo correspondiente.

- Modifico la columna **CodigoPostal** perteneciente a la tabla **TIENDA** con la instrucción **ALTER TIENDA TABLE MODIFY**. Le añado una restricción llamada **CP_UK** que no permitirá valores duplicados en la columna mediante la instrucción **CONSTRAINT CP_UK UNIQUE**



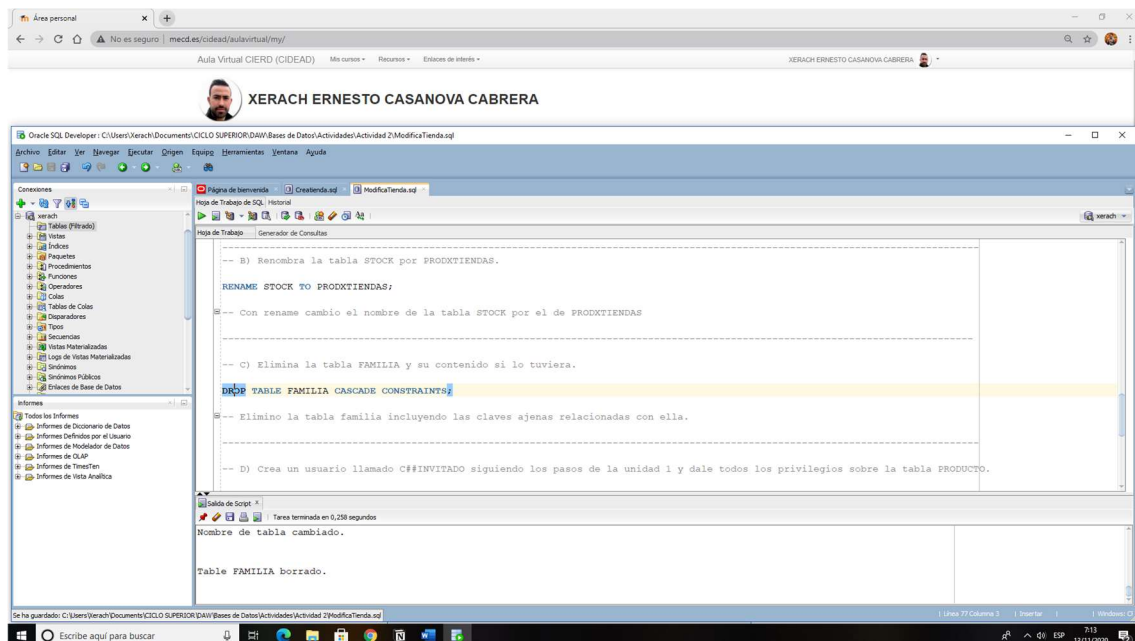
B) Renombra la tabla **STOCK** por **PRODXTIENDAS**.

- Utilizo la instrucción **RENAME STOCK TO PRODXTIENDAS**; para cambiarle el nombre a la tabla stock.



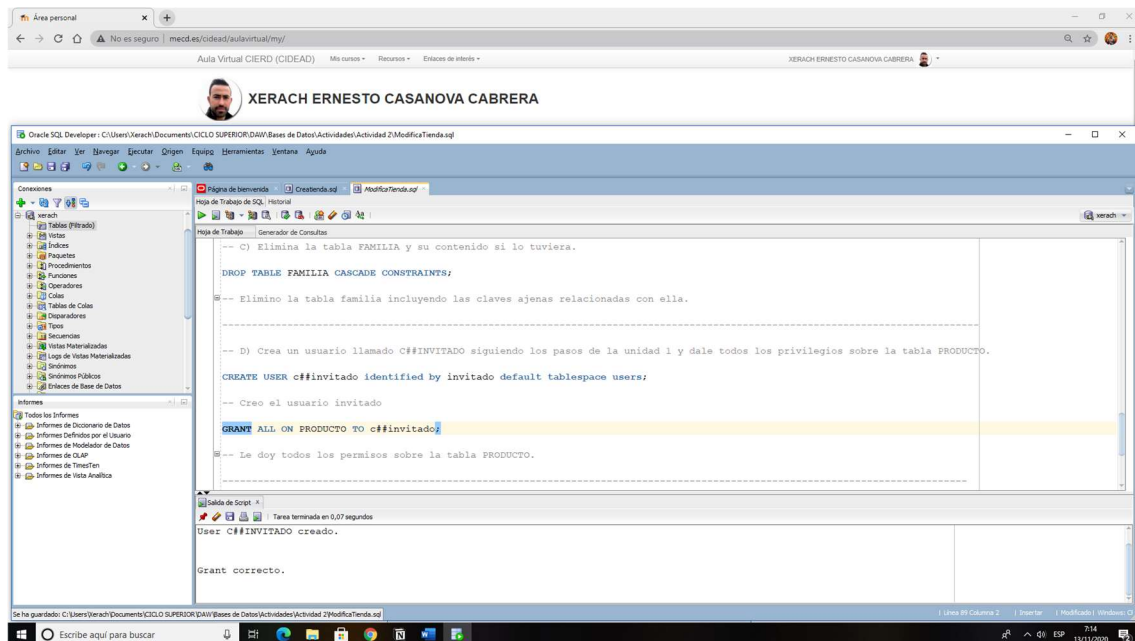
C) Elimina la tabla FAMILIA y su contenido si lo tuviera.

- Elimino la tabla FAMILIA con la instrucción ***DROP TABLE FAMILIA***, para que no de error debo indicarle que elimine también las claves ajenas en las tablas a las que hace referencia mediante la instrucción ***CASCADE CONSTRAINTS***;



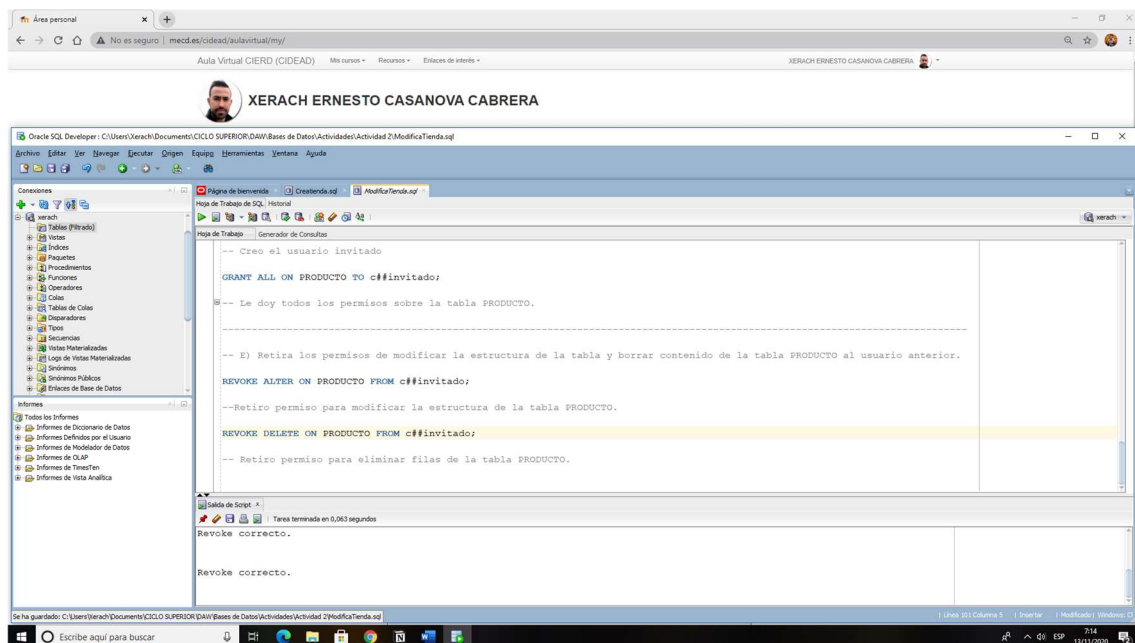
D) Crea un usuario llamado C##INVITADO siguiendo los pasos de la unidad 1 y dale todos los privilegios sobre la tabla PRODUCTO.

- Creo el usuario invitado con la instrucción ***CREATE USER C##INVITADO*** (le asigno una contraseña sencilla).
- Le asigno todos los permisos ***GRANT ALL*** sobre la tabla PRODUCTO.



E) Retira los permisos de modificar la estructura de la tabla y borrar contenido de la tabla PRODUCTO al usuario anterior.

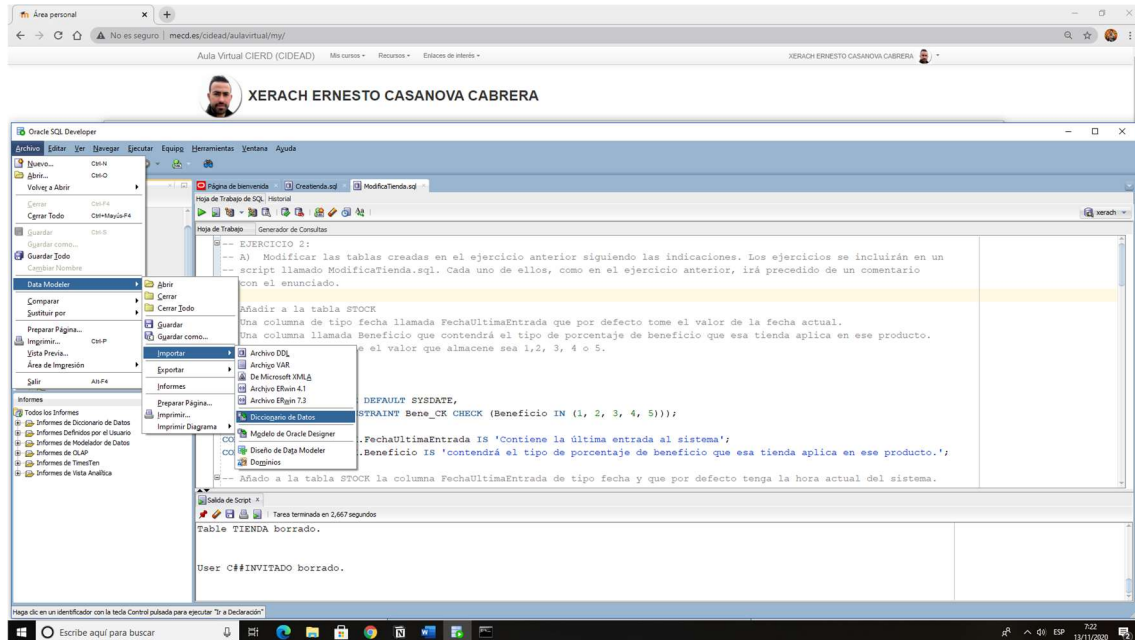
- Utilizo la instrucción **REVOKE ALTER** para retirar permisos de modificar la estructura de la tabla PRODUCTO.
- Con la instrucción **REVOKE DELETE** retiro permisos para borrar cualquier contenido que haya en la tabla.



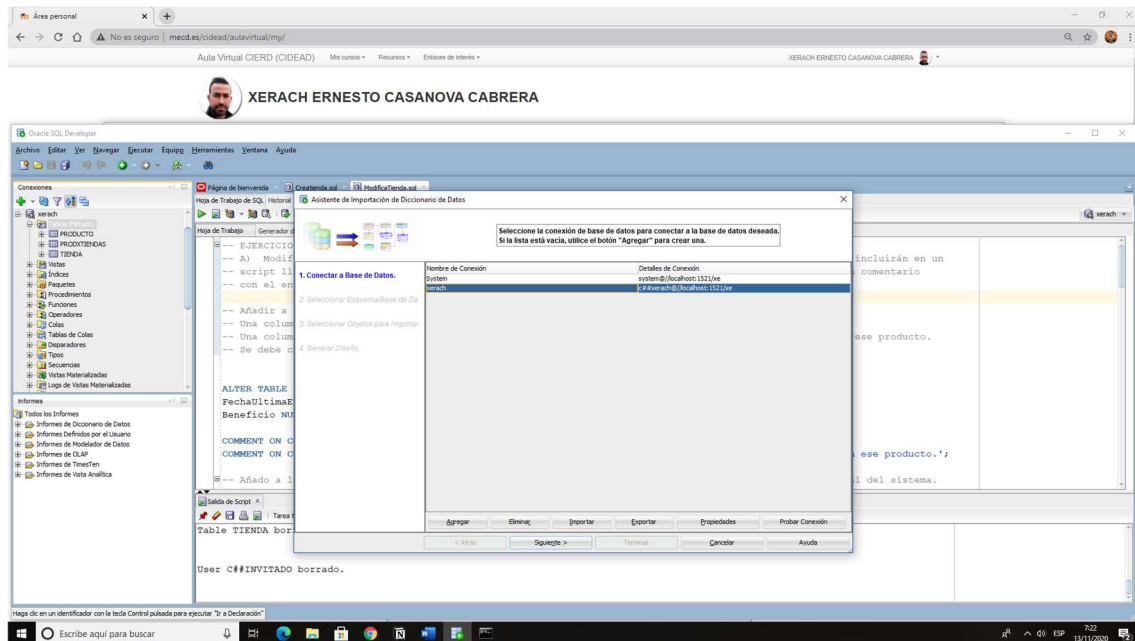
Ejercicio 3:

SQLDeveloper permite obtener el diagrama del modelo entidad relación a partir de las tablas ya creadas con la información contenida en el Diccionario de Datos. Una vez tengas realizados los ejercicios 1 y 2 genera el diagrama entidad relación y expórtalo en formato PNG.

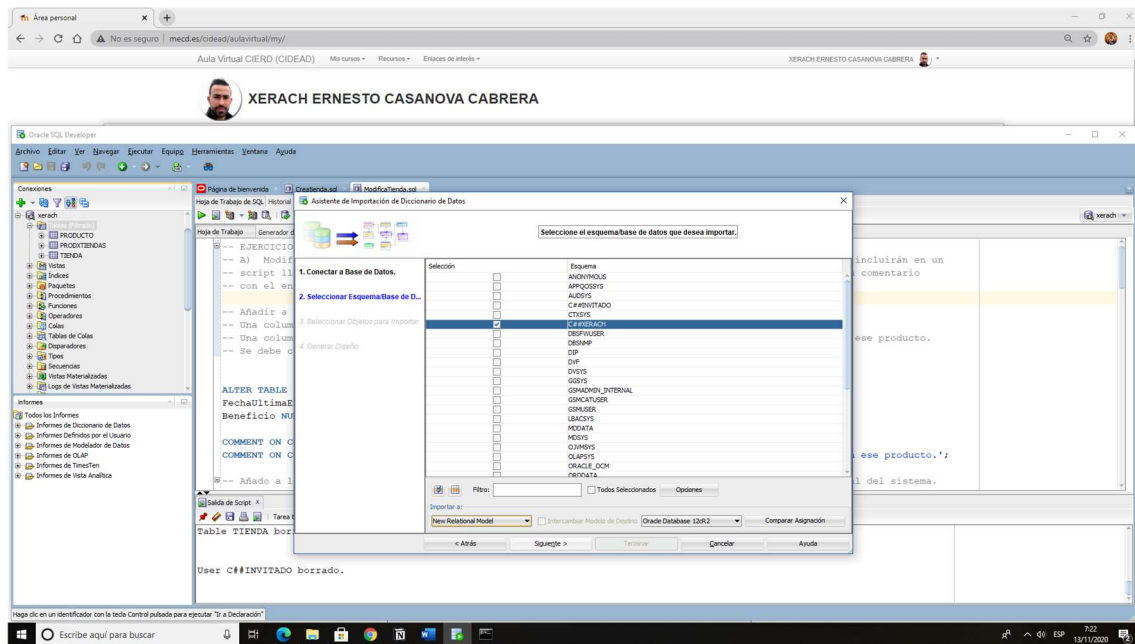
- Me dirijo a **Archivo/Data Modeler/Importar/Diccionario de Datos**



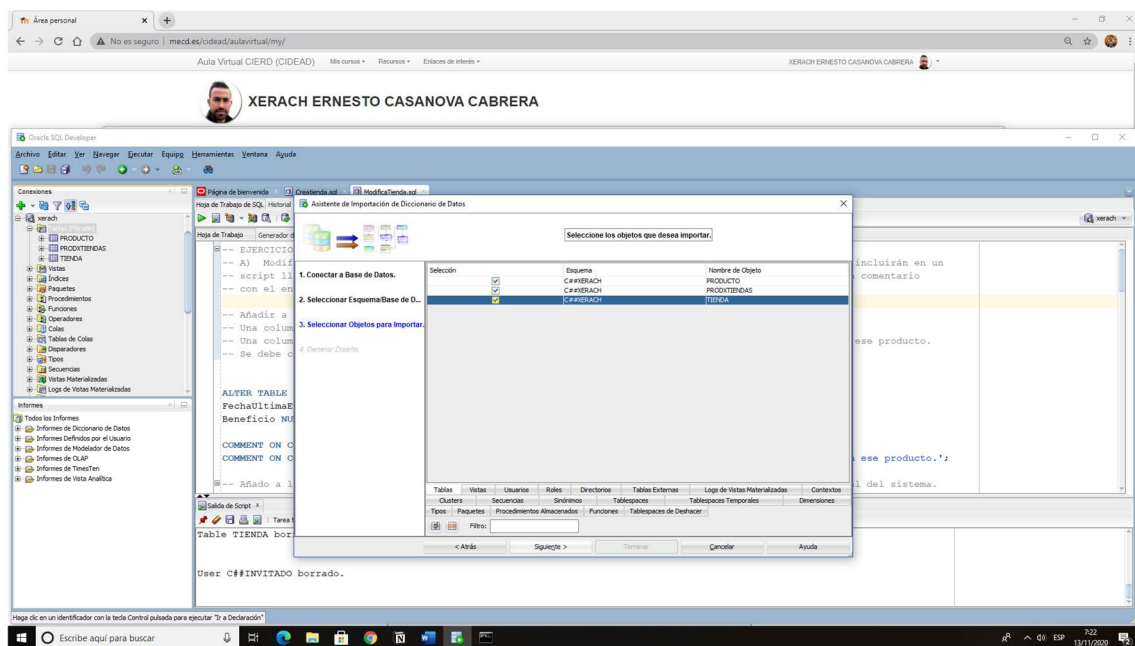
- Elijo mi nombre de usuario y le doy a siguiente



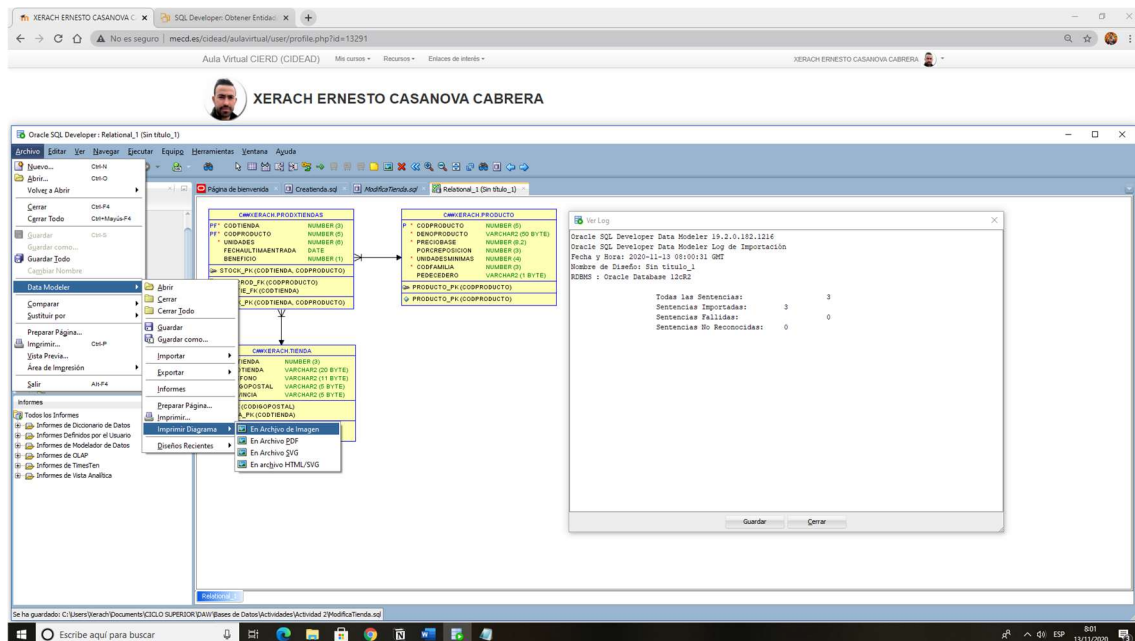
- En la siguiente pantalla elijo el esquema c##xerach y continúo



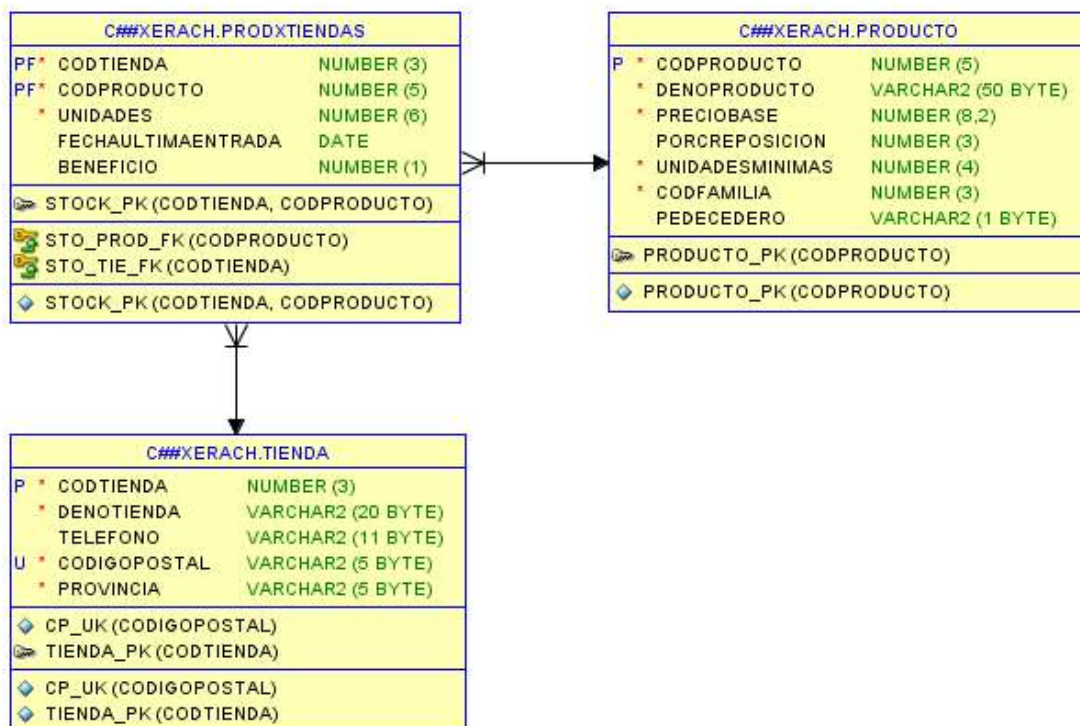
- Selecciono las tablas que deseo importar a la relación.
-



- Genero el diseño pulsando en terminar.



- Resultado:



Una vez realizado los dos Scripts, he procedido a borrar todas las tablas del ejercicio y he ejecutado los dos scripts en sqlplus, comprobando que se ejecutan perfectamente:

