

TAREA 2 DAW BASES DE DATOS

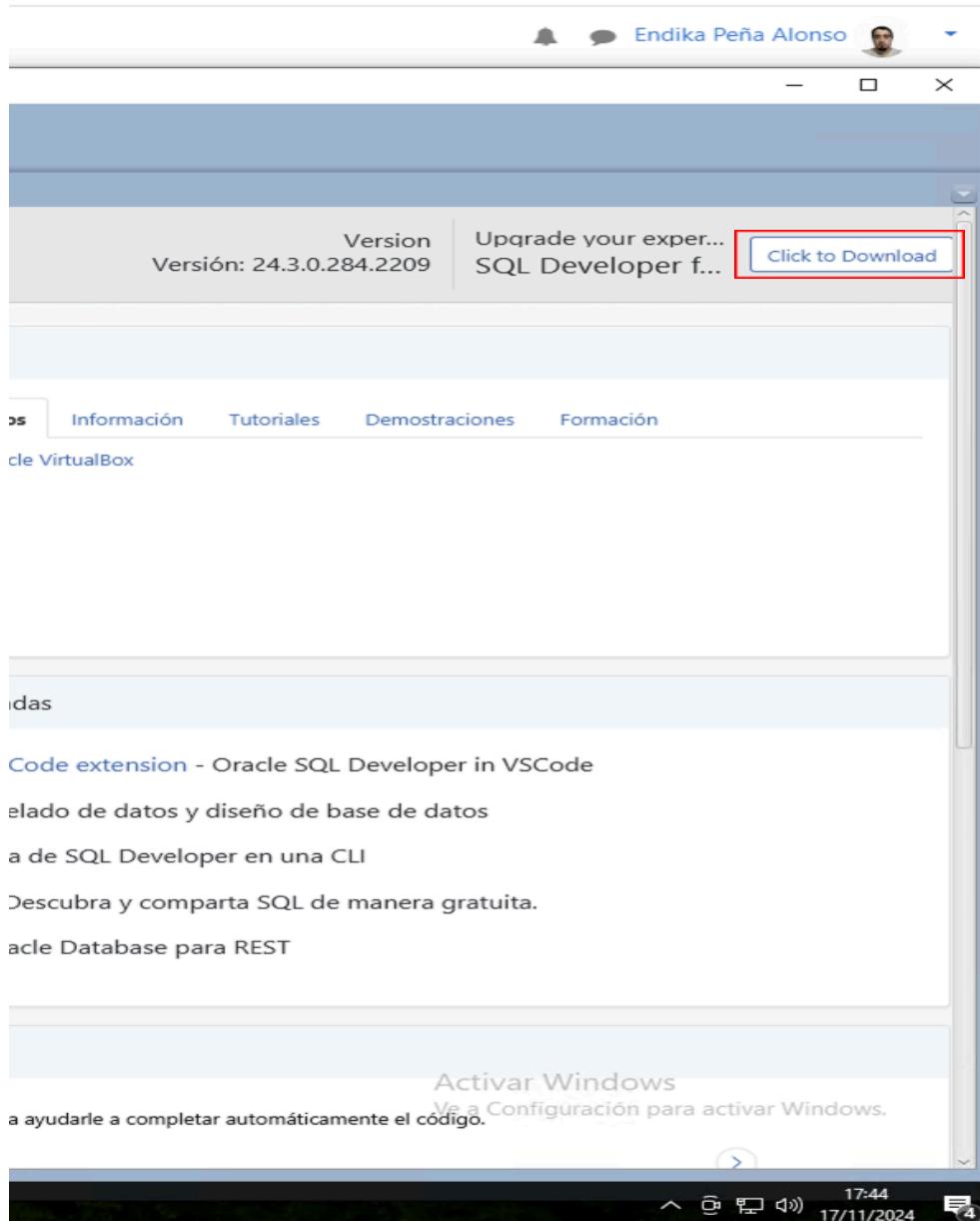
ENDIKA PEÑA ALONSO

Índice

Índice.....	2
Antes de empezar.....	3
Ejercicio 1: Creación de una base de datos de una red social.....	6
Tabla Usuarios:.....	6
Tabla Publicaciones:.....	6
Tabla Comentarios:.....	6
Tabla Amigos:.....	7
Tabla Reacciones:.....	7
Resumen de sentencias SQL que serán ejecutadas.....	8
Salida de la ejecución:.....	9
Visualizado desde SQL Developer “classic”:.....	10
Ejercicio 2: Modificar la base de datos anterior.....	11
Apartado A: Añadir una nueva tabla Mensajes:.....	11
Apartado B: Modificar la estructura de las tablas existentes.....	12
Cambiar el nombre de la tabla Amigos a Relaciones:.....	12
Cambiar el nombre de la columna fecha_registro en la tabla Usuarios a fecha_creacion:.....	13
Resumen.....	14
Ejercicio 3: Creación de usuarios - repaso.....	15
Apartado A: Crea un nuevo usuario invitado como vimos en la unidad 1 (recuerda añadir c##invitado por ser un usuario local) y dale todos los permisos sobre la tabla PUBLICACIONES.....	15
Apartado B: Retira los permisos de modificar la estructura de la tabla y borrar contenido de la tabla PUBLICACIONES al usuario invitado.....	17
Ejercicio 4: Investigación.....	18

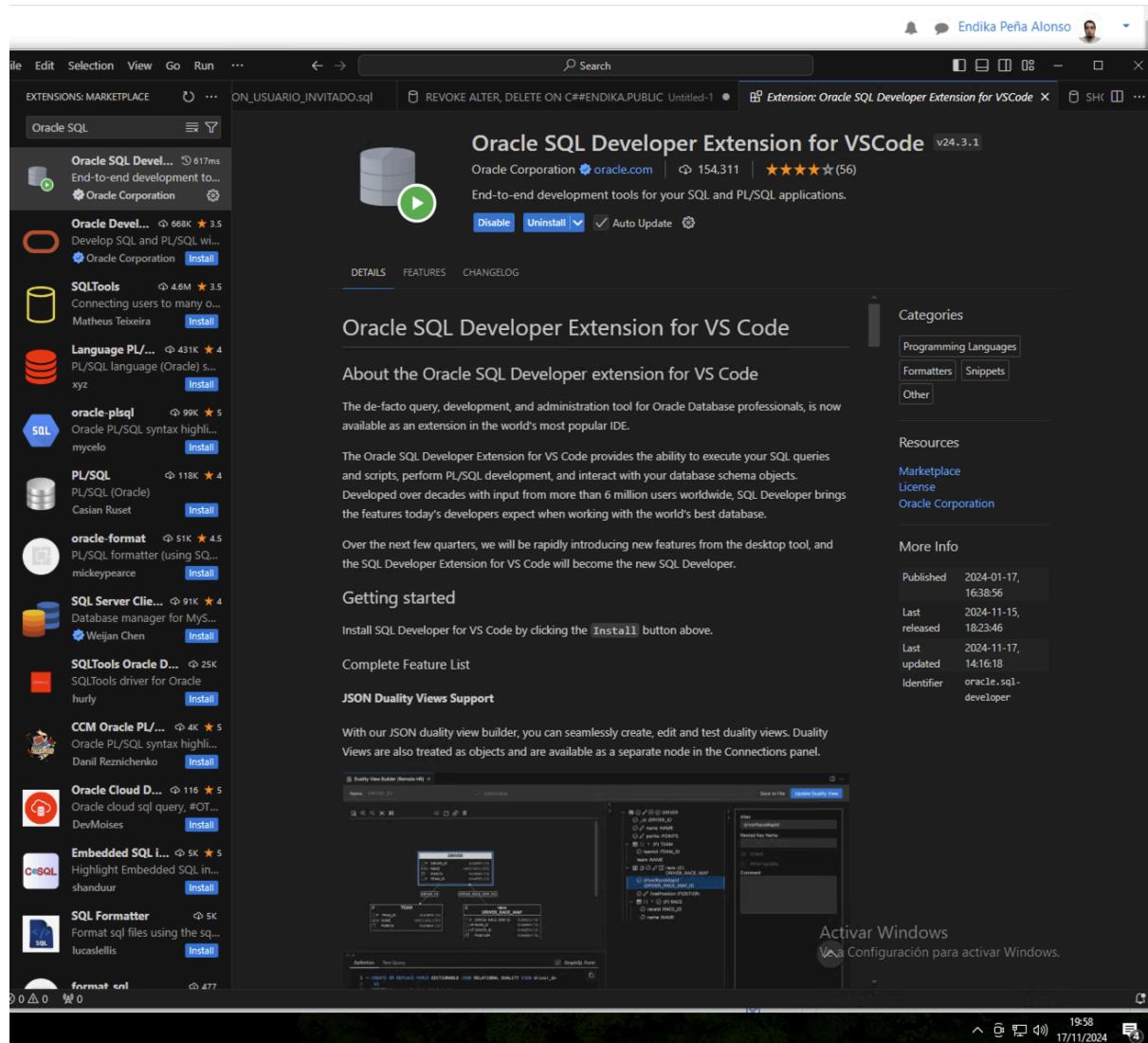
Antes de empezar

El ejercicio se ha realizado usando SQL Developer que durante su uso, aparece un botón para actualizarlo en la parte superior derecha del programa.

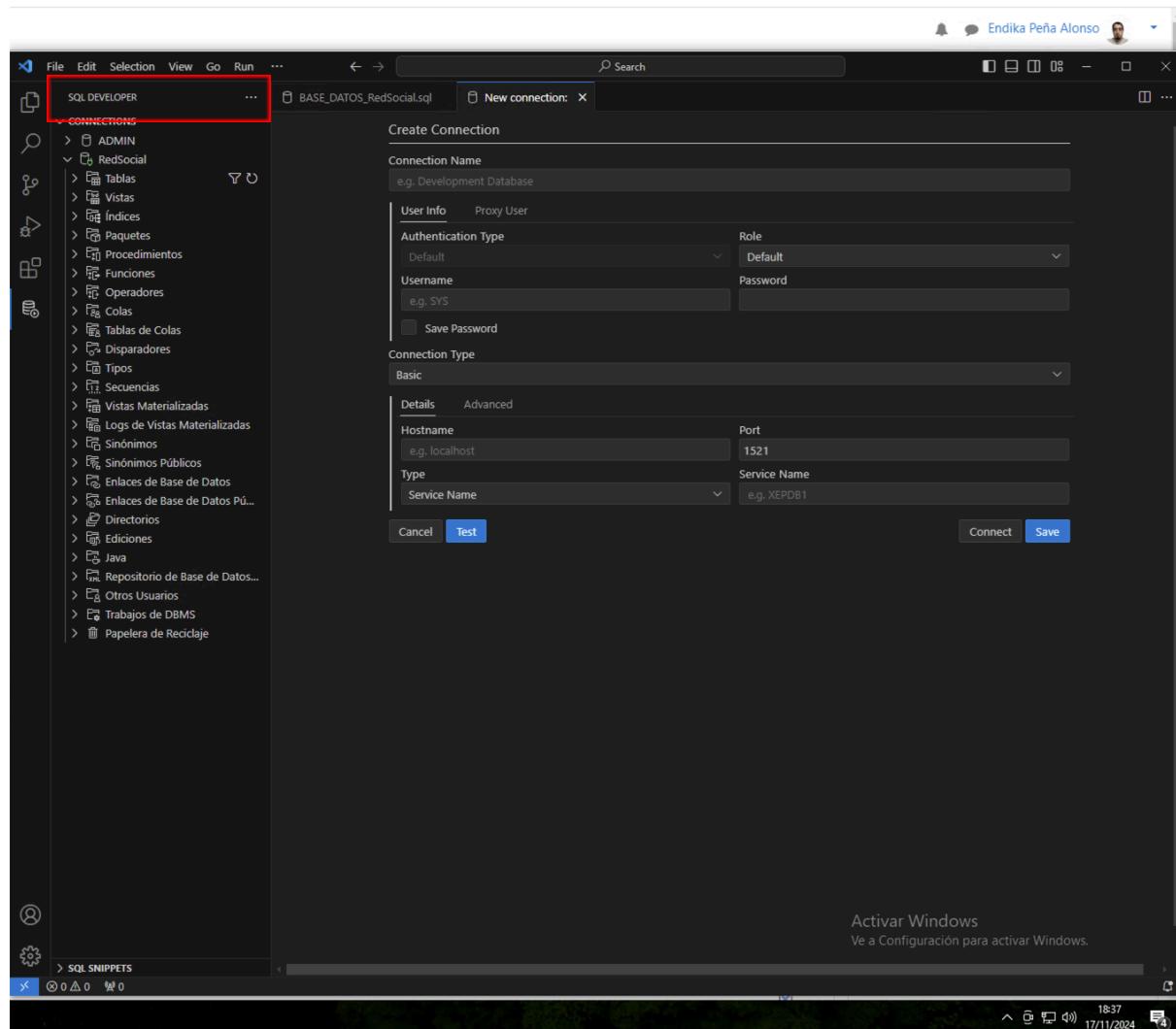


Al darle nos indica que existe otra versión de la herramienta como plugin o extensión de Visual Studio Code, que es la herramienta que he usado para realizar la práctica puesto que me ha parecido bastante más ligera, si bien la mayoría de funciones empleadas son las mismas.

Instalación SQLDeveloper en VSCode



A partir de ese momento he usado la extensión desde VSCode.



Ejercicio 1: Creación de una base de datos de una red social

Se te pide que diseñas la estructura de una base de datos para una red social simple. Esta red social permite a los usuarios registrarse, hacer publicaciones, comentar las publicaciones de otros usuarios, y formar relaciones de amistad. Sigue los siguientes pasos para crear las tablas necesarias.

Tabla Usuarios:

```
CREATE TABLE usuarios (
    id_usuario NUMBER GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1),
    nombre VARCHAR2(100) NOT NULL,
    email VARCHAR2(100) UNIQUE NOT NULL,
    contrasenia VARCHAR2(255) NOT NULL,
    fecha_registro DATE DEFAULT SYSDATE,
    foto_perfil VARCHAR2(255) NULL,
    CONSTRAINT usuarios_pk PRIMARY KEY(id_usuario)
)
```

Tabla Publicaciones:

```
CREATE TABLE publicaciones (
    id_publicacion NUMBER GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1),
    id_usuario NUMBER NOT NULL,
    contenido LONG NOT NULL,
    fecha_publicacion DATE DEFAULT SYSDATE,
    CONSTRAINT publicaciones_pk PRIMARY KEY(id_publicacion),
    CONSTRAINT usuarios_fk FOREIGN KEY(id_usuario) REFERENCES usuarios (id_usuario)
)
```

Tabla Comentarios:

```
CREATE TABLE comentarios (
    id_comentario NUMBER GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1),
    id_publicacion NUMBER NOT NULL,
    id_usuario NUMBER NOT NULL,
    comentario VARCHAR2(500) NOT NULL,
    fecha_comentario DATE DEFAULT SYSDATE,
    CONSTRAINT comentarios_pk PRIMARY KEY(id_comentario),
    CONSTRAINT comentarios_publicaciones_fk FOREIGN KEY(id_publicacion) REFERENCES publicaciones (id_publicacion),
    CONSTRAINT comentarios_usuarios_fk FOREIGN KEY(id_usuario) REFERENCES usuarios (id_usuario)
)
```

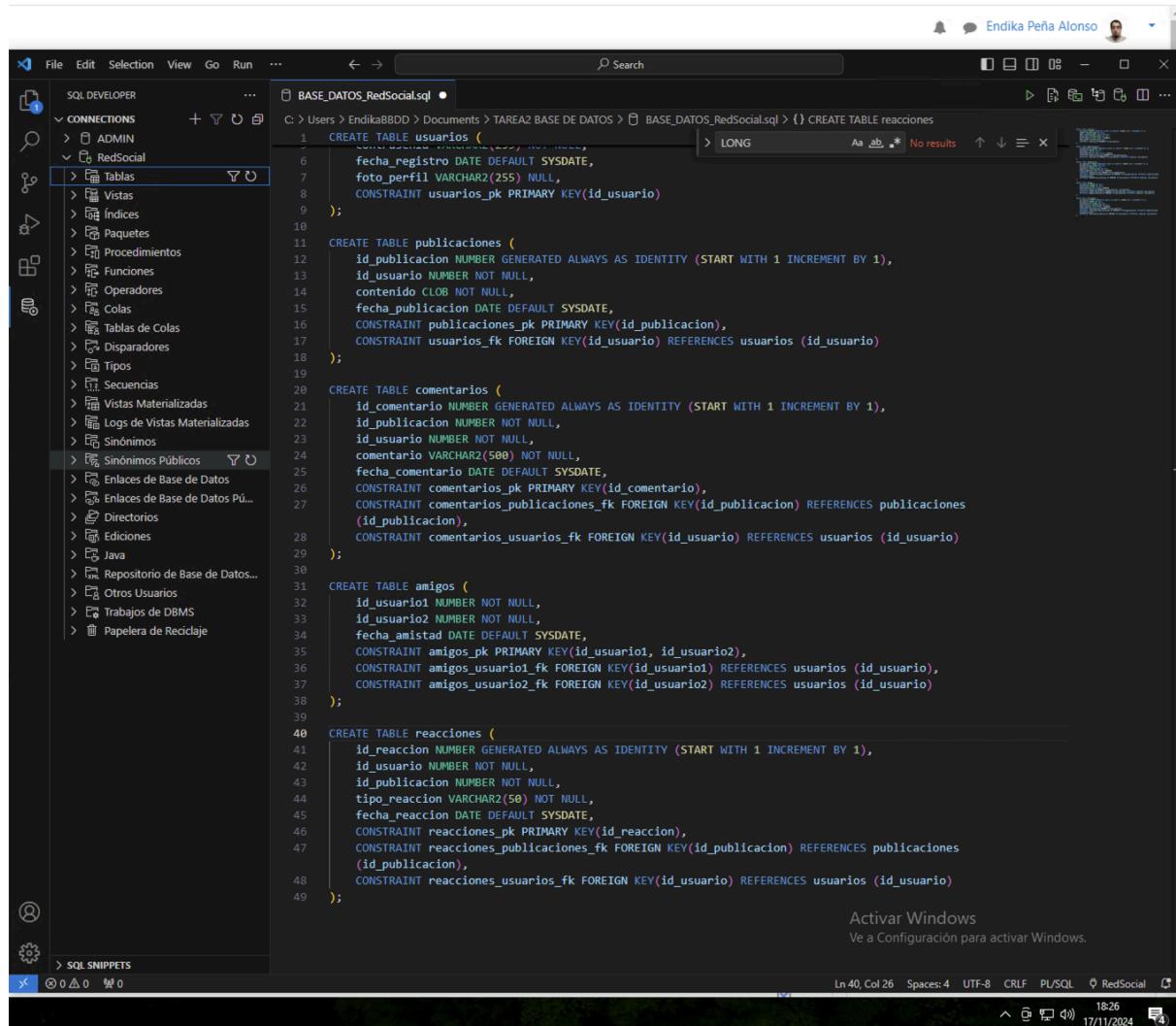
Tabla Amigos:

```
CREATE TABLE amigos (
    id_usuario1 NUMBER NOT NULL,
    id_usuario2 NUMBER NOT NULL,
    fecha_amistad DATE DEFAULT SYSDATE,
    CONSTRAINT amigos_pk PRIMARY KEY(id_usuario1, id_usuario2),
    CONSTRAINT amigos_usuario1_fk FOREIGN KEY(id_usuario1) REFERENCES usuarios (id_usuario),
    CONSTRAINT amigos_usuario1_fk FOREIGN KEY(id_usuario1) REFERENCES usuarios (id_usuario)
)
```

Tabla Reacciones:

```
CREATE TABLE reacciones (
    id_reaccion NUMBER GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1),
    id_usuario NUMBER NOT NULL,
    id_publicacion NUMBER NOT NULL,
    tipo_reaccion VARCHAR2(50) NOT NULL,
    fecha_reaccion DATE DEFAULT SYSDATE,
    CONSTRAINT reacciones_pk PRIMARY KEY(id_reaccion),
    CONSTRAINT reacciones_publicaciones_fk FOREIGN KEY(id_publicacion) REFERENCES publicaciones (id_publicacion),
    CONSTRAINT reacciones_usuarios_fk FOREIGN KEY(id_usuario) REFERENCES usuarios (id_usuario)
)
```

Resumen de sentencias SQL que serán ejecutadas.



The screenshot shows the Oracle SQL Developer interface. On the left, the Object Navigator pane displays various database objects under the connection 'RedSocial'. The main area shows a script named 'BASE_DATOS_RedSocial.sql' containing the following SQL code:

```
C:\> Users > EndikaBDD > Documents > TAREA2 BASE DE DATOS > BASE_DATOS_RedSocial.sql > () CREATE TABLE reacciones
1 CREATE TABLE usuarios (
2     id_usuario NUMBER GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1),
3     nombre VARCHAR2(50) NOT NULL,
4     apellido VARCHAR2(50) NOT NULL,
5     fecha_registro DATE DEFAULT SYSDATE,
6     foto_perfil VARCHAR2(255) NULL,
7     CONSTRAINT usuarios_pk PRIMARY KEY(id_usuario)
8 );
9 ;
10
11 CREATE TABLE publicaciones (
12     id_publicacion NUMBER GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1),
13     id_usuario NUMBER NOT NULL,
14     contenido CLOB NOT NULL,
15     fecha_publicacion DATE DEFAULT SYSDATE,
16     CONSTRAINT publicaciones_pk PRIMARY KEY(id_publicacion),
17     CONSTRAINT usuarios_fk FOREIGN KEY(id_usuario) REFERENCES usuarios (id_usuario)
18 );
19
20 CREATE TABLE comentarios (
21     id_comentario NUMBER GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1),
22     id_publicacion NUMBER NOT NULL,
23     id_usuario NUMBER NOT NULL,
24     comentario VARCHAR2(500) NOT NULL,
25     fecha_comentario DATE DEFAULT SYSDATE,
26     CONSTRAINT comentarios_pk PRIMARY KEY(id_comentario),
27     CONSTRAINT comentarios_publicaciones_fk FOREIGN KEY(id_publicacion) REFERENCES publicaciones
28     (id_publicacion),
29     CONSTRAINT comentarios_usuarios_fk FOREIGN KEY(id_usuario) REFERENCES usuarios (id_usuario)
30 );
31
32 CREATE TABLE amigos (
33     id_usuario1 NUMBER NOT NULL,
34     id_usuario2 NUMBER NOT NULL,
35     fecha_amistad DATE DEFAULT SYSDATE,
36     CONSTRAINT amigos_pk PRIMARY KEY(id_usuario1, id_usuario2),
37     CONSTRAINT amigos_usuario1_fk FOREIGN KEY(id_usuario1) REFERENCES usuarios (id_usuario),
38     CONSTRAINT amigos_usuario2_fk FOREIGN KEY(id_usuario2) REFERENCES usuarios (id_usuario)
39 );
40
41 CREATE TABLE reacciones (
42     id_reaccion NUMBER GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1),
43     id_usuario NUMBER NOT NULL,
44     id_publicacion NUMBER NOT NULL,
45     tipo_reaccion VARCHAR2(50) NOT NULL,
46     fecha_reaccion DATE DEFAULT SYSDATE,
47     CONSTRAINT reacciones_pk PRIMARY KEY(id_reaccion),
48     CONSTRAINT reacciones_publicaciones_fk FOREIGN KEY(id_publicacion) REFERENCES publicaciones
49     (id_publicacion),
50     CONSTRAINT reacciones_usuarios_fk FOREIGN KEY(id_usuario) REFERENCES usuarios (id_usuario)
```

The status bar at the bottom right shows the date and time: 17/11/2024 18:26.

Salida de la ejecución:

The screenshot shows the Oracle SQL Developer interface. The left sidebar displays the connection tree under 'CONNECTIONS' for 'RedSocial'. The main area shows the script 'BASE_DATOS_RedSocial.sql' with the following SQL code:

```
1 CREATE TABLE usuarios (
2     id_usuario NUMBER GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1),
3     nombre VARCHAR2(100) NOT NULL,
4     email VARCHAR2(100) UNIQUE NOT NULL,
5     contrasenia VARCHAR2(255) NOT NULL,
6     fecha_registro DATE DEFAULT SYSDATE,
7     foto_perfil VARCHAR2(255) NULL,
8     CONSTRAINT usuarios_pk PRIMARY KEY(id_usuario)
9 );
10
11 CREATE TABLE publicaciones (
12     id_publicacion NUMBER GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1),
13     id_usuario NUMBER NOT NULL,
14     contenido CLOB NOT NULL,
15     fecha_publicacion DATE DEFAULT SYSDATE,
16     CONSTRAINT publicaciones_pk PRIMARY KEY(id_publicacion),
17     CONSTRAINT usuarios_fk FOREIGN KEY(id_usuario) REFERENCES usuarios (id_usuario)
18 );
19
20 CREATE TABLE comentarios (
21     id_comentario NUMBER GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1),
22     id_publicacion NUMBER NOT NULL,
23     id_usuario NUMBER NOT NULL,
24     comentario VARCHAR2(500) NOT NULL,

```

The 'SCRIPT OUTPUT' tab is selected, showing the results of the executed commands:

```
Table USUARIOS created.

Table PUBLICACIONES created.

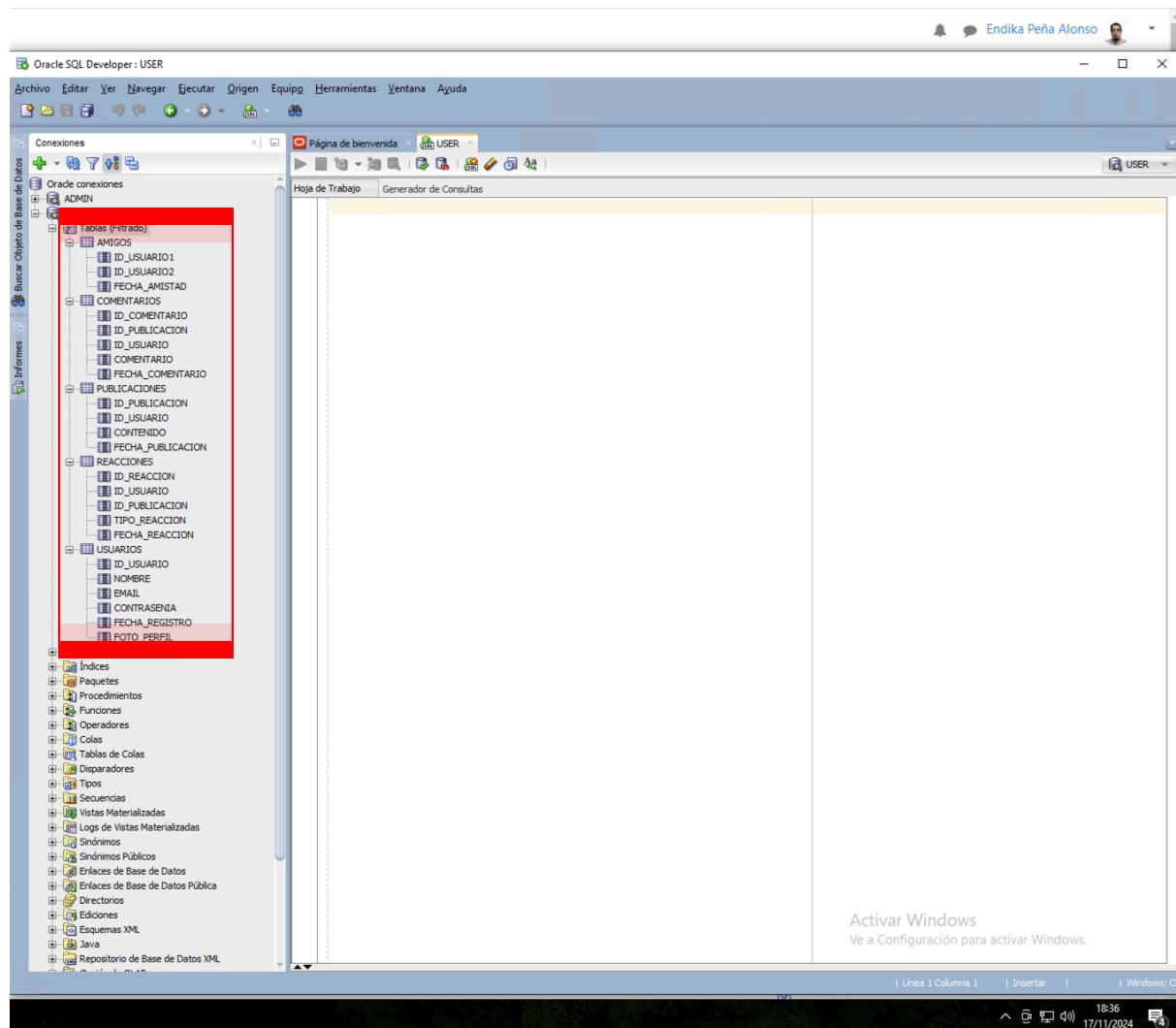
Table COMENTARIOS created.

Table AMIGOS created.

Table REACCIONES created.
```

At the bottom right, there is a message: 'Activar Windows' with the sub-instruction 'Ve a Configuración para activar Windows.'

Visualizado desde SQL Developer “classic”:



Ejercicio 2: Modificar la base de datos anterior

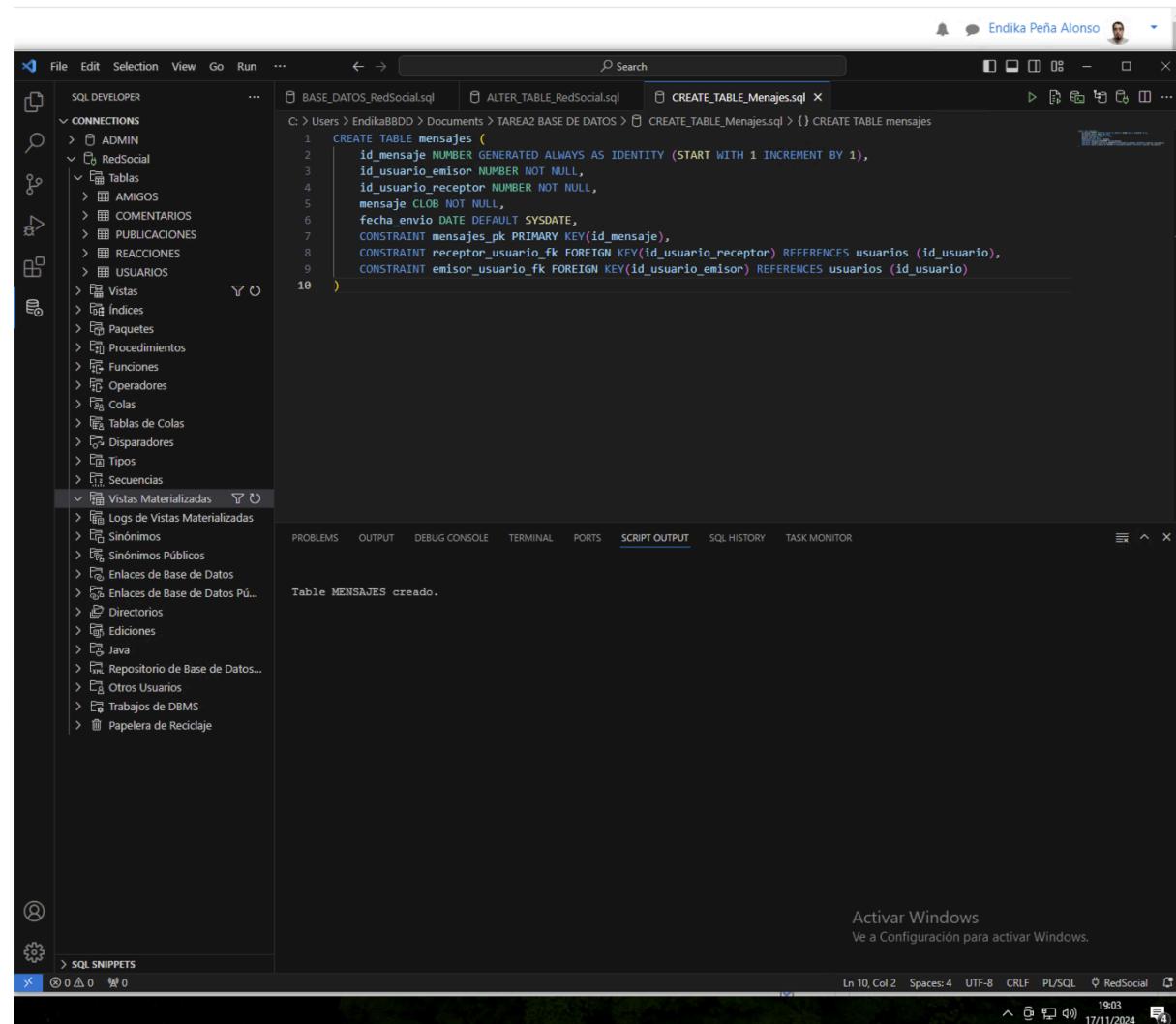
En esta etapa de la práctica vamos a modificar la base de datos del ejercicio anterior para adaptarnos mejor a las necesidades.

Apartado A: Añadir una nueva tabla Mensajes:

Enunciado:

Cada mensaje debe tener la siguiente estructura:

- `id_mensaje`: Identificador único del mensaje (clave primaria).
- `id_usuario_emisor`: Usuario que envió el mensaje (clave foránea que hace referencia a la tabla `Usuarios`).
- `id_usuario_receptor`: Usuario que recibió el mensaje (clave foránea que hace referencia a la tabla `Usuarios`).
- `mensaje`: Texto del mensaje.
- `fecha_envio`: Fecha y hora en que se envió el mensaje.



The screenshot shows the Oracle SQL Developer interface. On the left, the Object Navigator pane displays various database objects like connections, tables, and procedures. In the center, the Script Editor pane contains the SQL code for creating the 'mensajes' table:

```
CREATE TABLE mensajes (
    id_mensaje NUMBER GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1),
    id_usuario_emisor NUMBER NOT NULL,
    id_usuario_receptor NUMBER NOT NULL,
    mensaje CLOB NOT NULL,
    fecha_envio DATE DEFAULT SYSDATE,
    CONSTRAINT mensajes_pk PRIMARY KEY(id_mensaje),
    CONSTRAINT receptor_usuario_fk FOREIGN KEY(id_usuario_receptor) REFERENCES usuarios (id_usuario),
    CONSTRAINT emisor_usuario_fk FOREIGN KEY(id_usuario_emisor) REFERENCES usuarios (id_usuario)
);
```

Below the code, the message "Table MENSAJES created." is displayed. The bottom status bar shows the script is at line 10, column 2, with 1902 rows affected at 17/11/2024 19:02.

Apartado B: Modificar la estructura de las tablas existentes

En este apartado vamos a modificar ciertos aspectos de las tablas que ya existen, como el nombre de la misma o el nombre de algún atributo que tengan.

Cambiar el nombre de la tabla Amigos a Relaciones:

```
RENAME amigos TO relaciones;
```

The screenshot shows the Oracle SQL Developer interface. On the left is the Object Navigator pane, which lists various database objects under the RedSocial schema, including AMIGOS, COMENTARIOS, PUBLICACIONES, REACCIONES, USUARIOS, Vistas, Indices, Paquetes, Procedimientos, Funciones, Operadores, Colas, Tablas de Colas, Disparadores, Tipos, and Secuencias. Below this is a list of Materialized Views and their logs. The central workspace contains three tabs: 'BASE_DATOS_RedSocial.sql' (with the RENAME command), 'ALTER_TABLE_RedSocial.sql' (highlighted with a red dot), and 'CREATE_TABLE_Menajes.sql'. The status bar at the bottom indicates the script is in 'SCRIPT OUTPUT' mode, with 19:10 as the timestamp and 17/11/2024 as the date. A message 'Nombre de tabla cambiado.' (Table name changed) is displayed in the workspace.

Cambiar el nombre de la columna fecha_registro en la tabla Usuarios a fecha_creacion:

```
ALTER TABLE usuarios RENAME COLUMN fecha_registro TO fecha_creacion;
```

The screenshot shows the Oracle SQL Developer interface. On the left is the Object Navigator pane, which lists various database objects like ADMIN, RedSocial, and several tables such as AMIGOS, COMENTARIOS, PUBLICACIONES, REACCIONES, and USUARIOS. The central workspace contains a tab for 'ALTER_TABLE_Usuarios.sql' with the following content:

```
1  ALTER TABLE usuarios RENAME COLUMN fecha_registro TO fecha_creacion;
```

Below the code, the output window shows the message: "Table USUARIOS alterado." (Table USUARIOS altered). At the bottom of the interface, there are status bars for 'Activar Windows' (Activate Windows), connection information (Ln 1, Col 69, Spaces: 4, UTF-8, CRLF, PL/SQL, RedSocial), and system details (19:15, 17/11/2024).

Resumen

En este ejercicio se han realizado las siguientes acciones:

- Creación de una nueva tabla Mensajes.
- Modificación de la tabla Amigos a Relaciones (Esto lo podemos hacer porque aún no se está haciendo uso de ella, de lo contrario nos deberíamos coordinar con el equipo o equipos que hacen uso de la base de datos para consensuar cuando aplicar los cambios y con ello minimizar el impacto en una aplicación productiva)
- Modificación de un campo de la tabla Usuarios, fecha_registro a fecha_creacion.

The screenshot shows a database management interface with a sidebar on the left listing tables and their fields under the schema 'RedSocial'. Two SQL scripts are open in tabs at the top:

- Renombrando el nombre de la tabla**: Contains the command: `1 RENAME amigos TO relaciones;`
- Renombrando el campo de la tabla usuarios**: Contains the command: `1 ALTER TABLE usuarios RENAME COLUMN fecha_registro TO fecha_creacion;`

The bottom status bar indicates: 'Ln 3, Col 1 Spaces:4 UTF-8 CRLF PL/SQL No connection attached' and the date '17/11/2024'.

En la captura podemos ver las instrucciones ejecutadas y a la izquierda las tablas y sus respectivos campos.

Ejercicio 3: Creación de usuarios - repaso

En este ejercicio vamos a crear usuario y modificar permisos del usuario nuevo sobre una tabla concreta.

Apartado A: Crea un nuevo usuario invitado como vimos en la unidad 1 (recuerda añadir c##invitado por ser un usuario local) y dale todos los permisos sobre la tabla PUBLICACIONES.

Instrucciones para crear el usuario y otorgar permisos sobre la tabla publicaciones del usuario c##endika.

La capacidad de crear el usuario la tiene el administrador de la base de datos por tanto estas instrucciones SQL las debemos ejecutar con el.

The screenshot shows the Oracle SQL Developer interface. The left sidebar shows the connection tree under 'ADMIN' with 'Tablas' selected. The main pane displays the SQL script 'CREACION_USUARIO_INVITADO.sql' containing the following code:

```
1 -- Creación del nuevo usuario invitado
2 CREATE USER C##INVITADO IDENTIFIED BY 12345678;
3
4 -- Concedemos permisos para conectar
5 GRANT CONNECT TO C##INVITADO;
6
7 -- Concedemos todos los permisos sobre la tabla publicaciones del usuario c##endika
8 GRANT ALL ON C##ENDIKA.PUBLICACIONES TO C##INVITADO;
```

The 'SCRIPT OUTPUT' tab at the bottom shows the results of the execution:

```
User C##INVITADO creado.

Grant correcto.

Grant correcto.
```

At the bottom right, there is a message: "Activar Windows Ve a Configuración para activar Windows."

Comprobamos que el usuario C##INVITADO puede conectar y ver esa tabla.

The screenshot shows the SQL Developer interface. On the left, the Connections tree is expanded to show 'User invited' with various database objects like Tables, Views, Indices, Packages, Procedures, Functions, Operators, Queues, Queue Tables, Triggers, Sequences, Materialized Views, Log of Materialized Views, Synonyms, Public Synonyms, Data Links, Directories, Editions, Java, Repository, Other Users, Jobs, and Recycle Bin. The main area has two tabs open: 'CREACION_USUARIO_INVITADO.sql' and 'SHOW USER; Untitled-1'. The 'SHOW USER;' tab contains the following SQL code:

```
1 SHOW USER;
2 SELECT * FROM C##ENDIKA.PUBLICACIONES;
3 DESCRIBE C##ENDIKA.PUBLICACIONES;
```

The output of the 'SHOW USER;' command shows:

```
SQLCL: Versión 24.3 Production en dom nov 17 19:52:09 2024
Copyright (c) 1982, 2024, Oracle. Todos los derechos reservados.

Conectado a:
Oracle Database 21c Express Edition Release 21.0.0.0 - Production
Version 21.3.0.0.0

USER es "C##INVITADO"

no se ha seleccionado ninguna fila
```

The output of the 'DESCRIBE C##ENDIKA.PUBLICACIONES;' command shows the structure of the PUBLICACION table:

Nombre	NULL?	Tipo
ID_PUBLICACION	NOT NULL	NUMBER
ID_USUARIO	NOT NULL	NUMBER
CONTENIDO	NOT NULL	CLOB
FECHA_PUBLICACION		DATE

At the bottom, the status bar shows: Ln 1, Col 11 Spaces:4 UTF-8 CRLF PL/SQL User invited 19:52 17/11/2024

Apartado B: Retira los permisos de modificar la estructura de la tabla y borrar contenido de la tabla PUBLICACIONES al usuario invitado.

```
REVOKE ALTER, DELETE ON C##ENDIKA.PUBLICACIONES FROM C##INVITADO;
```

Repasamos la salida de ejecutar dicho comando.

The screenshot shows the Oracle SQL Developer interface. On the left, the 'CONNECTIONS' tree view is expanded, showing various database objects under the 'ADMIN' connection. In the center, there are two tabs: 'CREACION_USUARIO_INVITADO.sql' and 'REVOKE ALTER, DELETE ON C##ENDIKA.PUBLIC Untitled-1'. The second tab contains the SQL command: 'REVOKE ALTER, DELETE ON C##ENDIKA.PUBLICACIONES FROM C##INVITADO;'. Below the tabs, the 'SCRIPT OUTPUT' tab is selected, displaying the message 'Revoke correcto.' (Revoke successful). At the bottom of the screen, status information includes 'Ln 1, Col 66', 'Spaces: 4', 'UTF-8', 'CRLF', 'PL/SQL', 'ADMIN', '19:55', '17/11/2024', and a file icon.

Ejercicio 4: Investigación

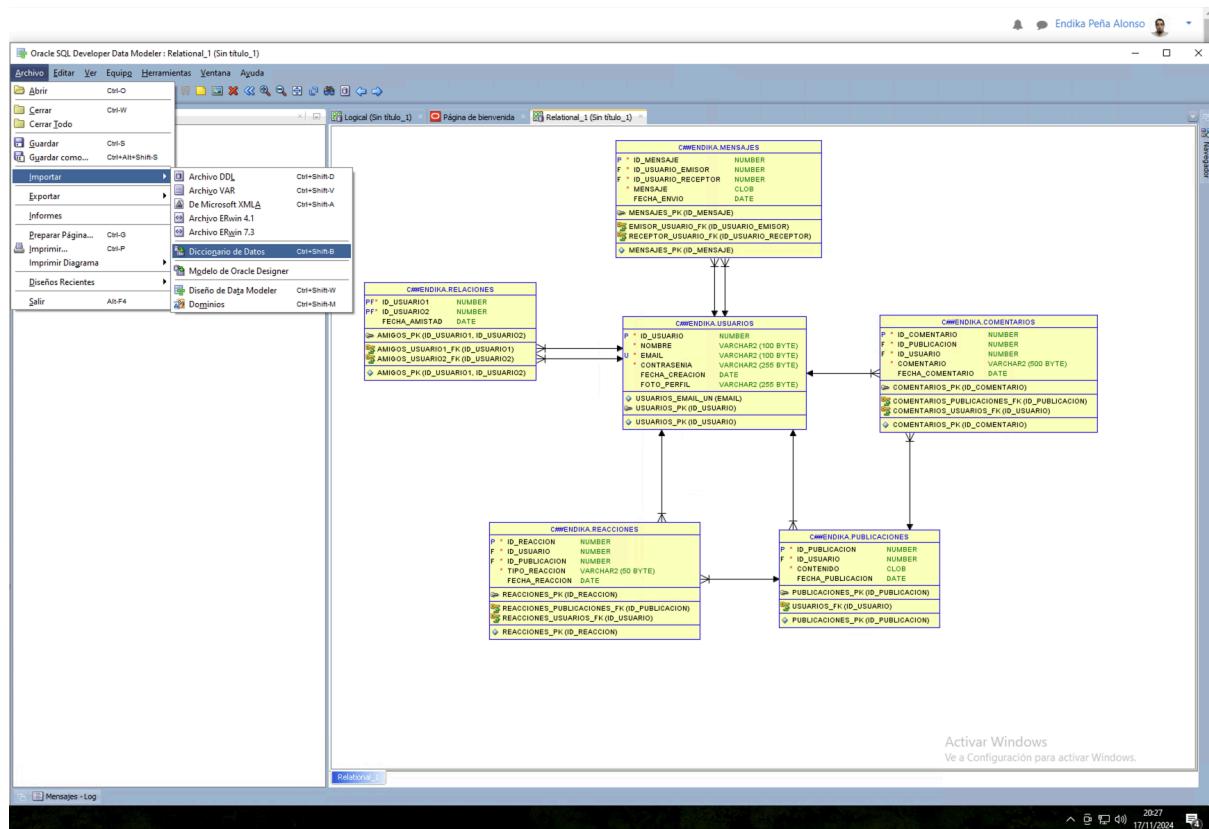
Enunciado

SQL DEVELOPER Y SQL DATA MODELER

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.

La tarea a entregar es la captura de las tablas del modelo relacional de nuestra base de datos red social ordenadas para que puedan verse las diferentes relaciones entre las tablas.

Para realizar esta parte he tenido que instalar el programa de SQL DATA MODELER, en él he tenido que importar desde un diccionario de datos, luego he seleccionado las tablas que quiero importar y se ha generado el diagrama.



Una vez el diagrama ha sido ordenado el resultado es el siguiente:

