



Grado en Ingeniería Informática

APLICACIONES DE BASES DE DATOS

SESION 1

Entorno Oracle

Docentes:

Raúl Marticorena

Jesús Maudes



Índice de contenidos

1. INTRODUCCIÓN	3
2. OBJETIVOS	3
3. GUIÓN DE PRÁCTICAS	3
3.1 Arranque y parada de servicios en Windows	3
3.2 Arranque y parada de servicios en GNU/Linux (UBUNTU)	4
3.3 Arranque y parada desde el cliente SQL*Plus	4
3.4 Breve guía de comandos en SQL*Plus	5
Ejercicios	8
3.5 SQLDeveloper	8
Ejercicio	13
3.6 Algunas diferencias con PostgreSQL	13
3.7 Oracle Application Express (APEX)	14
Ejercicio	20
4. RESUMEN	20
5. GLOSARIO	20
6. RECURSOS	20



1. Introducción

El siguiente guión de prácticas presenta el uso básico del entorno y herramientas utilizados en la asignatura, relativas al manejo del sistema gestor de bases de datos (SGBD) utilizado en la asignatura, Oracle.

Partiendo de una primera toma de contacto con el entorno de consola, SQL*Plus, para el acceso inicial a la base de datos Oracle. Posteriormente se presenta el uso de una herramienta gráfica como SQLDeveloper, y se finaliza con la introducción al entorno Oracle Application Express (APEX) para la generación de aplicaciones *web* con formularios e informes.

2. Objetivos

- Conocer las herramientas fundamentales para interactuar con el SGBD Oracle, tanto en consola con SQL*Plus como en modo gráfico con SQLDeveloper.
- Tomar un primer contacto con la herramienta de generación de formularios e informes APEX incluida en Oracle.

3. Guión de prácticas

A continuación se detallan los pasos a seguir por el alumno, que debe intentar repetir dichos pasos a modo de guión. Al final de cada sección se plantean algunos ejercicios para su resolución por parte del alumno.

3.1 Arranque y parada de servicios en Windows

Si los servicios de Oracle, en particular **OracleServiceXE** (ojo, no la instancia y base de datos) no están arrancados recibiremos un mensaje de error similar al realizar cualquier intento de conexión con la base de datos:

ERROR:

ORA-12560: TNS:protocol adapter error

En un **S.O. Windows**, cuando arrancamos con la opción *Start Database* en el menú de Oracle (abrir el menú *Inicio* del sistema operativo, en *Todos los programas*) (*StartDB.bat* en *%ORACLE_HOME%\bin*) se lanzan dos servicios en el script correspondiente (*OracleXETNSListener* y *OracleServiceXE*) que deben estar funcionando. En el *script* adicionalmente se arranca de la instancia de bases de datos con nombre XE (acrónimo de *eXpress Edition*).

La configuración por defecto arranca estos dos servicios, por lo que la base de datos está siempre funcionando desde el inicio (si se ha instalado correctamente, aunque puede tardar un tiempo).

Hasta que no ejecutemos la opción *Stop Database* (*StopDB.bat* en *%ORACLE_HOME%\bin*) en el menú mencionado previamente, no se para el servicio *OracleServiceXE*. Sin embargo *OracleXETNSListener* se deja funcionando y podemos ver su estado desde línea de comandos con:

```
$> lsnrctl status
```



El *listener* permite conexiones remotas (en el puerto 1521) pero con *OracleServiceXE* podemos realizar conexiones locales con el *listener* apagado.

3.2 Arranque y parada de servicios en GNU/Linux (UBUNTU)

Para comprobar si los servicios están arrancados debemos ejecutar:

```
$> sudo service oracle-xe status
```

Como resultado de la invocación se mostrarán una información similar a la siguiente con un círculo verde, indicando el estado activo (si está inactivo se muestra un punto blanco) y el arranque de los dos servicios equivalentes a los de Windows, uno de escucha de peticiones (*listener*) y otro de base de datos (instancia de la base de datos):

```
● oracle-xe.service - LSB: Oracle 11g Express Edition
   Loaded: loaded (/etc/init.d/oracle-xe; generated)
   Active: active (exited) since Sat 2020-01-11 14:51:16 CET; 2h 3min ago
     Docs: man:systemd-sysv-generator(8)
   Process: 671 ExecStart=/etc/init.d/oracle-xe start (code=exited, status=0/SUCC

ene 11 14:50:00 UBU systemd[1]: Starting LSB: Oracle 11g Express Edition...
ene 11 14:50:09 UBU oracle-xe[671]: Starting Oracle Net Listener.
ene 11 14:50:14 UBU su[792]: (to oracle) root on none
ene 11 14:50:15 UBU su[792]: pam_unix(su:session): session opened for user oracl
ene 11 14:50:32 UBU oracle-xe[671]: Starting Oracle Database 11g Express Edition
ene 11 14:50:32 UBU su[970]: (to oracle) root on none
ene 11 14:50:32 UBU su[970]: pam_unix(su:session): session opened for user oracl
ene 11 14:51:16 UBU systemd[1]: Started LSB: Oracle 11g Express Edition.
```

Si queremos detener los servicios ejecutamos

```
$>sudo service oracle-xe stop
```

Para iniciar los servicios ejecutamos

```
$>sudo service oracle-xe start
```

Para detener y reiniciar servicios, en un solo paso:

```
$>sudo service oracle-xe restart
```

En la instalación por defecto (disponible en la máquina virtual proporcionada), el servidor estará arrancado por defecto y no debería ser necesario usar inicialmente estas opciones de parada y arranque.

3.3 Arranque y parada desde el cliente SQL*Plus

Para probar la parada (*shutdown*) y el arranque (*startup*) manual del SGBD desde el cliente SQL*Plus se inicia sesión ejecutando desde una consola del sistema operativo¹:

¹ Se supone que usamos la máquina virtual, donde los usuarios `sys` y `system` tienen como contraseñas 1234, y el usuario `hr` tiene como contraseña `hr`.



Aplicaciones de Bases de Datos
Grado en Ingeniería Informática
\$> sqlplus sys/1234 as sysdba

Si todo funciona correctamente, entramos en la consola de SQL*Plus como administradores (el *prompt* del sistema habrá cambiado a `SQL>`), y a continuación paramos la base de datos.

```
SQL> SHUTDOWN IMMEDIATE
```

La parada de la instancia de base de datos **no tiene relación con la parada y arranque de los dos servicios** previamente mencionados (en Windows o GNU/Linux)

Para volver a arrancar la instancia de base de datos ejecutamos:

```
SQL> STARTUP
```

Cerramos sesión en SQL*Plus:

```
SQL> exit
```

con la instancia parada (habiendo ejecutado previamente `SQL>shutdown`), e intentamos entrar de nuevo

```
$> sqlplus sys/1234 as sysdba
```

el mensaje nos avisa de que la instancia está parada, aunque sí se admite la conexión.

```
Connected to an idle instance.
```

Es necesario arrancar de nuevo la instancia para empezar a trabajar.

```
SQL> STARTUP
```

3.4 Breve guía de comandos en SQL*Plus

Salimos de SQL*Plus (con `exit`) y probamos a conectar con otro usuario como `system` (la contraseña utilizada en la asignatura para este usuario también es `1234`).

```
$> sqlplus system
```

Para salir de la sesión iniciada tecleamos:

```
SQL> exit
```

Nota: si hemos realizado la instalación en Windows: para conectarse de nuevo a SQL*Plus validando contra usuario del SO del grupo `ORA_DBA`:

```
$> sqlplus / as sysdba
```

Se conecta por defecto en este caso con el usuario `SYS`. Para conectar manualmente como `SYS` siempre es obligatorio indicar el rol `SYSDBA` o `SYSOPER`. El usuario con el que hemos realizado la instalación se habrá añadido durante la instalación al grupo `ORA_DBA`. En este caso, se valida directamente contra la cuenta del sistema operativo.

Nota: si ejecutamos el comando en un **S.O Windows**, `lusrmgr.msc`, se abre la consola de administración de usuarios/grupos y podemos ver el grupo `ORA_DBA`, al que pertenece el usuario con el que hemos iniciado sesión en el sistema operativo. Dependiendo del usuario con el que hayamos iniciado la sesión en el sistema operativo, podremos conectarnos o no. Por ejemplo, si accedemos con una cuenta de invitado o cuenta de alumno podremos encontrarnos con problemas.

Un usuario típico para pruebas es el usuario `hr` (siglas de Human Resources) con password `hr`.

```
$> sqlplus hr/hr
```



Nota importante: si no consigues entrar con dicho usuario, típicamente el error se debe a que estamos ante un Oracle “recién instalado”, con el usuario `hr` bloqueado. Si fuese así:

1. Entra como administrador, por ejemplo como administrador *system* con *password 1234*:
`$> sqlplus system/1234`
2. Ejecutar el siguiente comando para desbloquearlo y asignar el password `hr`. Nota que el *password* sí que es sensible a mayúsculas/minúsculas, mientras el nombre del usuario no.

```
SQL> ALTER USER HR ACCOUNT UNLOCK IDENTIFIED BY hr;
```

Conectarse a la base de datos como usuario `SYSTEM` con rol `SYSDBA`.

Bien desde el sistema operativo.

```
$> sqlplus SYSTEM AS SYSDBA
```

Nota: Nos pedirá el *password* sin mostrarse en pantalla. Otra opción es `$>sqlplus SYSTEM/password as sysdba` pero se muestra el *password* en pantalla)

O bien entrando en SQL*Plus sin validación (`$> sqlplus /nolog`) y conectándose posteriormente:

```
SQL> CONNECT SYSTEM AS SYSDBA
```

Nota: se puede desconectar al usuario con `SQL>disconnect` o bien realizando una nueva conexión que implícitamente conlleva desconexión.

Para cambiar el *password* del usuario actual:

```
SQL> PASSWORD
```

Para conectar con el usuario `hr` (en modo local):

```
SQL> CONNECT HR/hr
```

Por defecto nos conecta al esquema (SCHEMA) `HR` con el usuario `HR` de la instancia `XE`. En Oracle el esquema se corresponde con el usuario y es donde se guardarán todos los objetos que creemos con dicho usuario (tablas, índices, procedimientos, funciones, *triggers*, etc.)

Comprobar esto con:

```
SQL> SELECT USER FROM DUAL;
```

`Dual` es una “tabla ficticia” en Oracle con distintas utilidades, entre ellas poder realizar consultas generales al sistema, como el usuario actual.

También se puede consultar el usuario actual de estas otras tres formas equivalentes:

```
SQL> show user
```

```
SQL> SELECT SYS_CONTEXT( 'USERENV', 'CURRENT_SCHEMA' ) FROM DUAL;
```

```
SQL> SELECT * FROM GLOBAL_NAME;
```

Nota: se puede cambiar a otro esquema con `ALTER SESSION SET CURRENT_SCHEMA=OTHER_SCHEMA_NAME;`

Mostrar las tablas disponibles del esquema por defecto:

```
SQL> SELECT TABLE_NAME FROM USER_TABLES;
```



Se debe mostrar una salida similar a:

```
TABLE_NAME
-----
REGIONS
LOCATIONS
DEPARTMENTS
JOBS
EMPLOYEES
JOB_HISTORY
COUNTRIES
```

7 rows selected.

Los comandos SQL, como el anterior, acaban con ";". Sin embargo hay otros comandos **no – SQL, propios de Oracle y SQL*Plus**, los cuáles **no necesitan terminar en ";"**. Por ejemplo, si queremos saber la estructura de la tabla `REGIONS`, podemos utilizar el comando `DESC` (ó `DESCRIBE`). Por ejemplo, para obtener una descripción de la tabla:

```
SQL> DESC REGIONS
```

Se pueden consultar los contenidos (datos) de las tablas del usuario `HR`, por ejemplo de su tabla `REGIONS`:

```
SQL> SELECT * FROM REGIONS;
```

Si no estamos en el esquema correspondiente (por ejemplo, hemos alterado la sesión para cambiar el esquema actual a `SYSTEM` o hemos conectado con otro usuario) es necesario indicar el nombre de esquema en la consulta previamente al nombre del objeto (tabla en este caso):

```
SQL> SELECT * FROM HR.REGIONS;
```

Para la ejecución de *scripts* desde el sistema operativo, con comandos desde SQL*PLUS se utiliza un carácter `@` previamente al nombre del script a a ejecutar:

```
SQL> @file
```

o

```
SQL> @path/file
```

Si no se especifica el *path* se busca en el **directorio actual** desde donde se arrancó SQL*Plus. Por convenio, los ficheros llevan extensión `.sql`. Si se utilizan directorios con rutas en blanco se deben incorporar dobles comillas para poder localizarse adecuadamente.

Para saber cuál es el directorio de arranque actual:

- En Windows con: `SQL> host cd`
- En GNU/Linux con: `SQL> host pwd`

Por ejemplo, para ejecutar un script en una ruta que contiene espacios en blanco ejecutaríamos:

```
SQL> @"ruta con espacios/en blanco/a un directorio/script.sql"
```

Se pueden definir variables con la opción `DEFINE`, para posteriormente referenciarlas con un simbolo `&` delante.

```
SQL> DEFINE var=ruta con espacios/en blanco/a un directorio
```



Si se quiere utilizar en la invocación del script se debe hacer de nuevo utilizando dobles comillas

```
SQL> @"&var\script.sql"
```

Para resolver el problema de las dobles comillas (ejemplo con rutas Windows):

```
SQL> DEFINE var ="C:\Documents and Settings\alumno"
```

```
SQL> @"&var\holamundol.sql"
```

Es necesario utilizar las dobles comillas dos veces.

También podemos ejecutar el *script* directamente desde línea de comandos del sistema operativo así:

```
$> sqlplus hr/hr @"\users\alumno\desktop\mi_consulta.sql"
```

Pero en este caso se observará que, quizás, tras ejecutar los comandos en el *script*, **NO** retorna al la consola del sistema operativo, sino que permanece registrado en la sesión de SQL*Plus. Para evitar esa situación, **recuerda incluir un comando EXIT** (sin punto y coma al final), al final del *script*.

Observa además, que hay que especificar la ruta al *script*. La única forma de evitar especificar la ruta es cambiar al directorio que contiene el *script* "antes" de entrar en SQL*Plus. Por ejemplo:

```
$> cd /users/alumno/desktop  
$> sqlplus hr/hr @mi_consulta.sql
```

Ejercicios

1. Ejecutar un *script* creado por el alumno, desde SQL*Plus, que nos muestre el usuario actual con el que estamos conectados.
2. Dibujar con cualquier herramienta de dibujo disponible (o con papel y boli), el diagrama entidad-relación correspondiente a la base de datos (esquema) HR que se puede deducir utilizando las utilidades vistas de SQL*Plus.

3.5 SQLDeveloper

SQLDeveloper es una herramienta gráfica para interactuar con Oracle. Pudiendo hacer básicamente lo mismo que desde SQL*Plus, introduce muchas utilidades adicionales que facilitan el manejo del SGBD. Para crear una conexión para el usuario HR en SQLDeveloper (equivalente a crear una conexión en SQL*Plus), desde la zona izquierda, en *Conexiones* botón derecho, *Nueva Conexión*.

NOTA: En la máquina virtual subida a UBU virtual está instalada la versión 19 de Sql Developer, para ciertas herramientas con el Data Modeler es conveniente actualizar a la versión 21. Ver instrucciones en ANEXO al final de este documento.



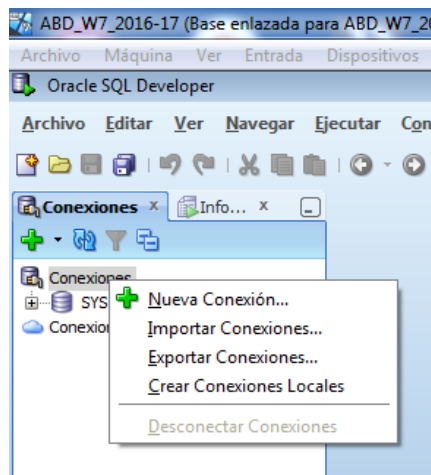


Ilustración 1: Creación de una nueva conexión en SQLDeveloper

Completamos los valores de conexión, dando un nombre a la conexión, e indicando usuario (`HR`) y contraseña (`hr`) al igual que en SQL*Plus dejando el resto de valores como están (valores por defecto).

Es muy importante respetar las mayúsculas/minúsculas a la hora de introducir el *password*. Si marcamos el checkbox "Guardar contraseña" almacena la misma, en caso contrario nos preguntará la contraseña cada vez que nos conectemos.

Es muy interesante probar la conexión mediante el botón "Probar" antes de dar al botón "Guardar". De esta forma, si hubiese un mensaje de error, como por ejemplo que el *password* o el usuario sea incorrecto, tenemos opción de cambiarlo. Un problema típico es que el usuario esté bloqueado porque la instalación esté recién hecha. La resolución de este problema ya viene detallada en la sección previa. Si no hubiera problemas, debería salirnos el mensaje "Estado: Correcto" que se ve en la esquina inferior izquierda (encima del botón "Ayuda") de la Ilustración 2.

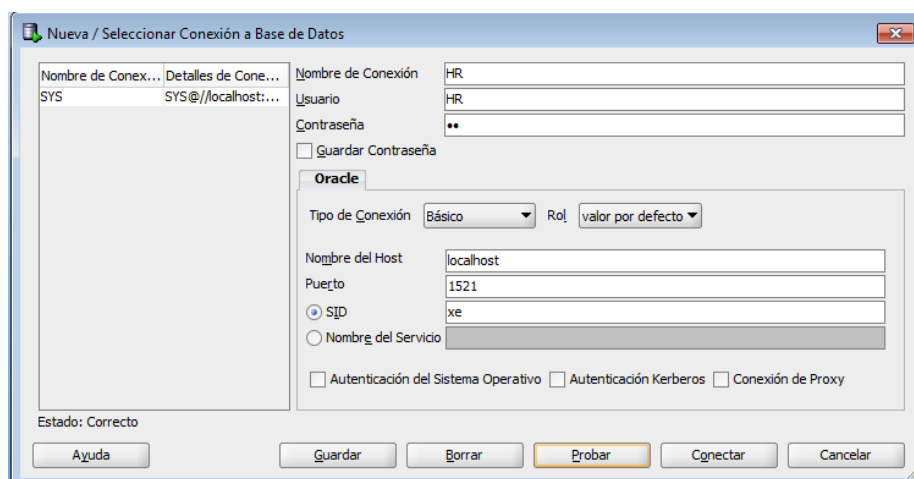


Ilustración 2: Configuración de una conexión

Si queremos crear una conexión para el usuario `SYS` o `SYSTEM` con rol `SYSDBA` se debe cambiar en el desplegable *Rol*, seleccionando dicho rol concreto:



Al pulsar al botón “*Guardar*” la conexión queda registrada de forma permanente y la podemos utilizar en sucesivas sesiones de SQLDeveloper. Al ser guardada, aparecerá en la solapa de conexiones, como muestra la Ilustración 3.

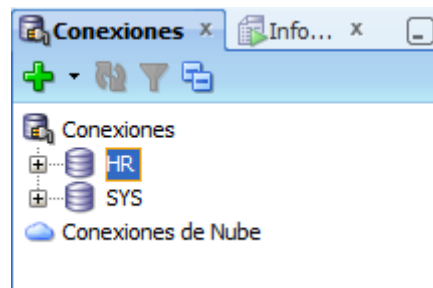


Ilustración 3: La conexión HR ya ha sido definida y guardada.

Ahora podemos seleccionar con el botón derecho del ratón, en dicha conexión y:

- Seleccionar *Propiedades* si queremos cambiar alguna de las propiedades definidas en la ventana de diálogo de la Ilustración 2.
- También podemos seleccionar la opción *Conectar*, que en caso de no haber guardado la contraseña nos la pedirá.

Una vez te hayas conectado te saldrá una “*hoja de trabajo*” en blanco con el nombre de la conexión, como se muestra en la Ilustración 4.

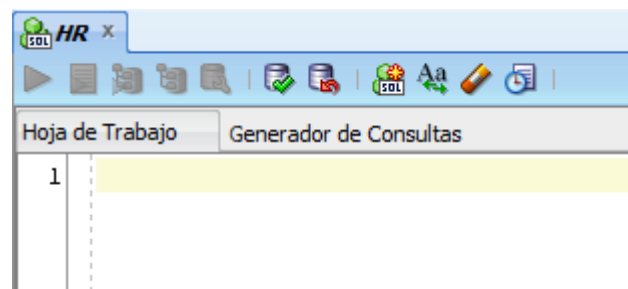


Ilustración 4: Hoja de Trabajo en blanco para la conexión HR.

Ahora podríamos escribir uno o varios comandos SQL en la hoja de trabajo, como en la Ilustración 5. En esta ilustración se ve como en la parte de la izquierda aparecen las 7 tablas del usuario HR. Si desplegásemos una cualquiera de las tablas podríamos ver la descripción de sus campos. No obstante, SQLDeveloper te ayuda sugiriéndote los nombres de tablas y campos a medida que vas escribiendo los comandos SQL.



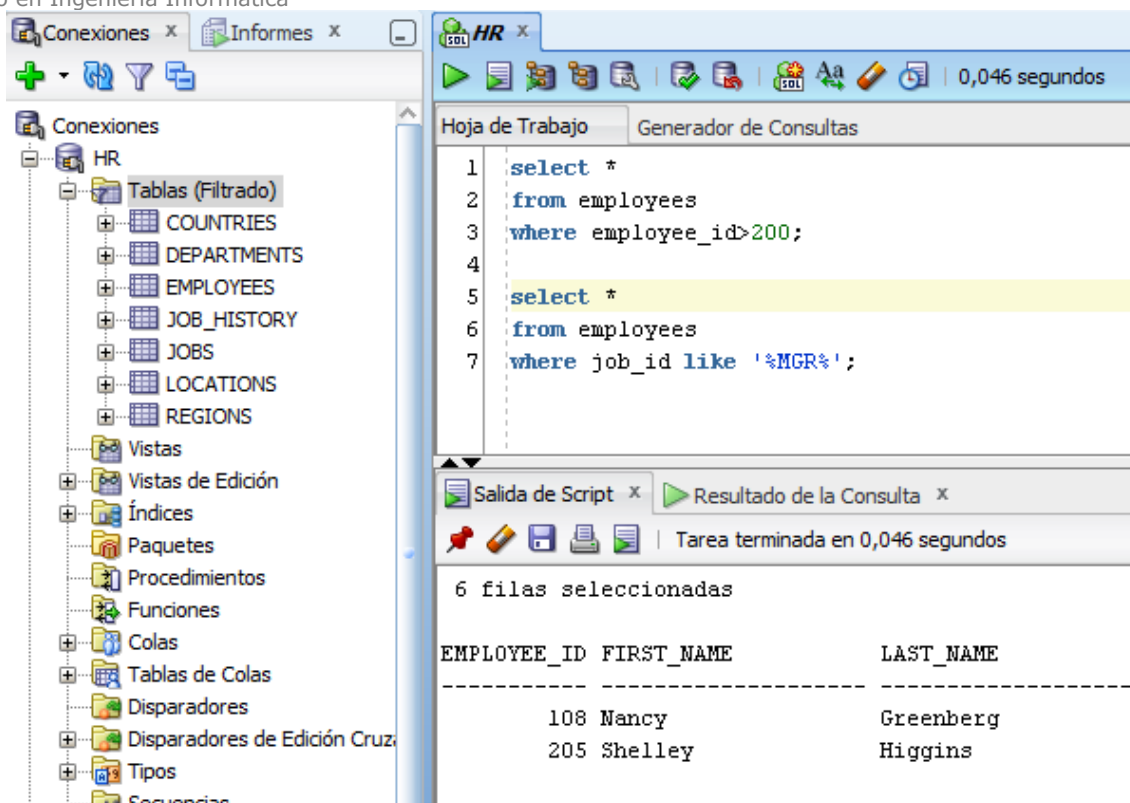




Ilustración 5: Ejecución de varios comandos SQL.

En la parte superior izquierda de la hoja de trabajo tienes dos botones muy importantes:

-  sirve para ejecutar únicamente el comando SQL sobre el que esté situado actualmente el cursor del editor de texto de la hoja de trabajo.
-  sirve para ejecutar secuencialmente todos los comandos SQL de la hoja de trabajo. Está pensado para que el contenido de la hoja de trabajo sea un *script* de comandos. De hecho, la ventana donde se puede ver el resultado de esa ejecución se llama "Salida del Script", como se puede apreciar en la Ilustración 5².

La salida del script puede borrarse con el botón  que aparece en dicha ventana "Salida del Script".

En la hoja de trabajo nos aparecen los números de línea. Esta opción no está por defecto pero resulta interesante, ya que en caso de que haya un error, SQLDeveloper nos informa de la línea donde se ha producido. Para que los números de línea sean visibles basta seleccionar con el botón derecho en la barra vertical que indica la Ilustración 6, y seleccionar la opción "Conmutar Números de Línea".

² Nota que una diferencia con el *pgAdmin* de *PostgreSQL*, es que en la salida de *SQLDeveloper* se ve el resultado de todas las SELECT, no sólo de la última.



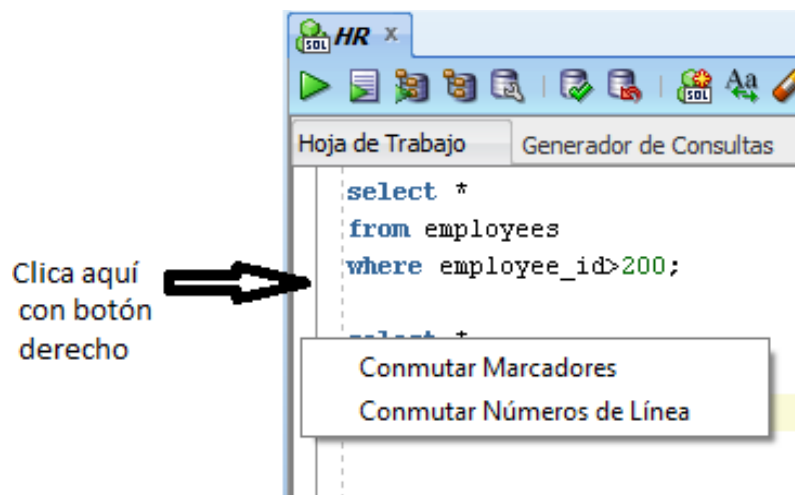





Ilustración 6: Activar la visibilidad de los números de línea.

Se pueden agregar más hojas de trabajo con el botón , que te permite especificar a qué conexión está asociada la nueva hoja (o con Alt+F10).

El contenido de una hoja de trabajo puede guardarse en forma de *script* SQL con el botón  o con *Archivo>Guardar* (como puedes editar varias hojas de trabajo simultáneamente, te guardaría la que estés editando en ese momento).

Puedes abrir un *script* con el botón  o con *Archivo > Abrir*, de manera que crea una hoja de trabajo nueva con el contenido del *script*, pudiendo editarlo y ejecutarlo como ya se ha descrito anteriormente. Tras ejecutar un *script*, el panel de la izquierda es posible que no refleje ningún cambio. Por ejemplo, es posible que hayamos creado o borrado tablas pero siguen apareciendo las mismas.

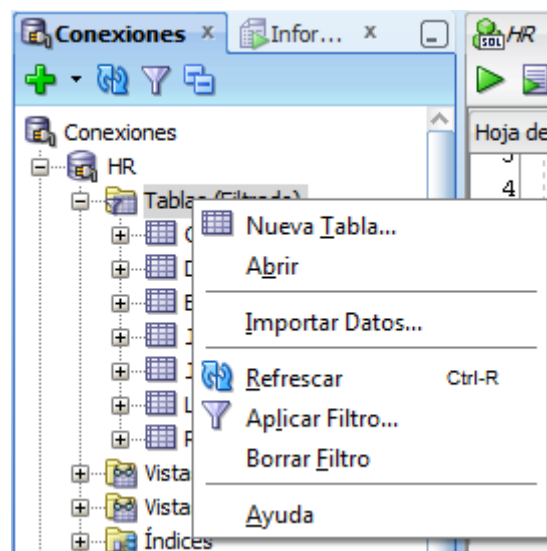



Ilustración 7: Refrescar el contenido del panel izquierdo.

Por ello, es necesario forzar el refresco de la información de ese panel. En la Ilustración 7 puede verse que hay un botón *refrescar*  tanto en la parte superior seleccionando el botón derecho en cada uno de los ítems de dicho panel. Por ejemplo, en la ilustración se ve como con el botón derecho estamos refrescando las tablas de la conexión HR.



Una vez creada la conexión también podemos, por ejemplo, visualizar las tablas disponibles, que deben coincidir con las vistas desde SQL*Plus, simplemente navegando en el árbol desplegable a la conexión y eligiendo *Tablas*.

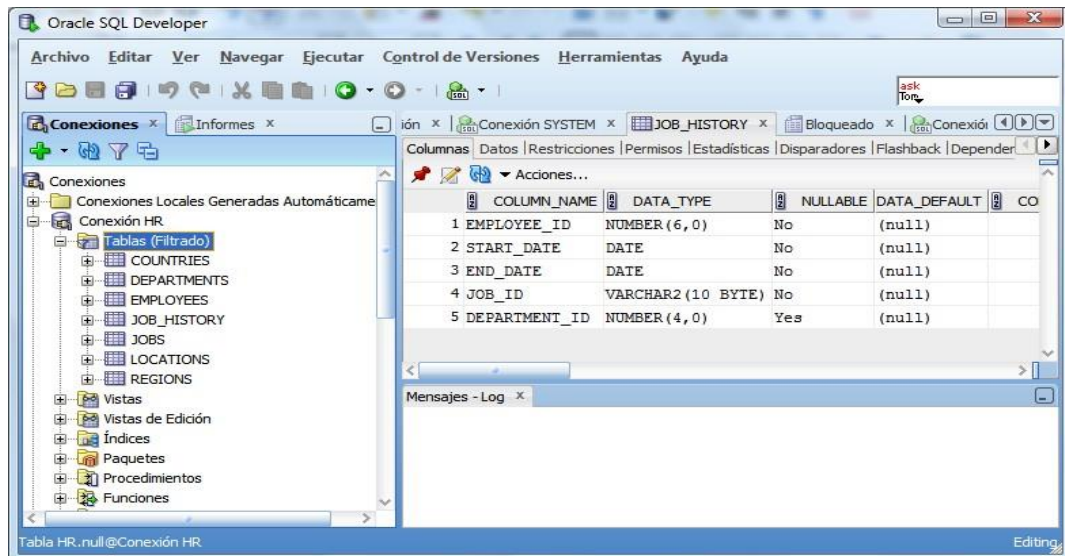


Ilustración 8: Pantalla principal de SQLDeveloper, visualización de tablas

En el menú *Archivo / Data Modeler / Importar / Diccionario de Datos*. Seleccionamos la conexión (Ej:HR), esquema (HR) y tablas (todas las tablas) a partir de los cuáles se puede realizar ingeniería inversa para generar el modelo relacional.

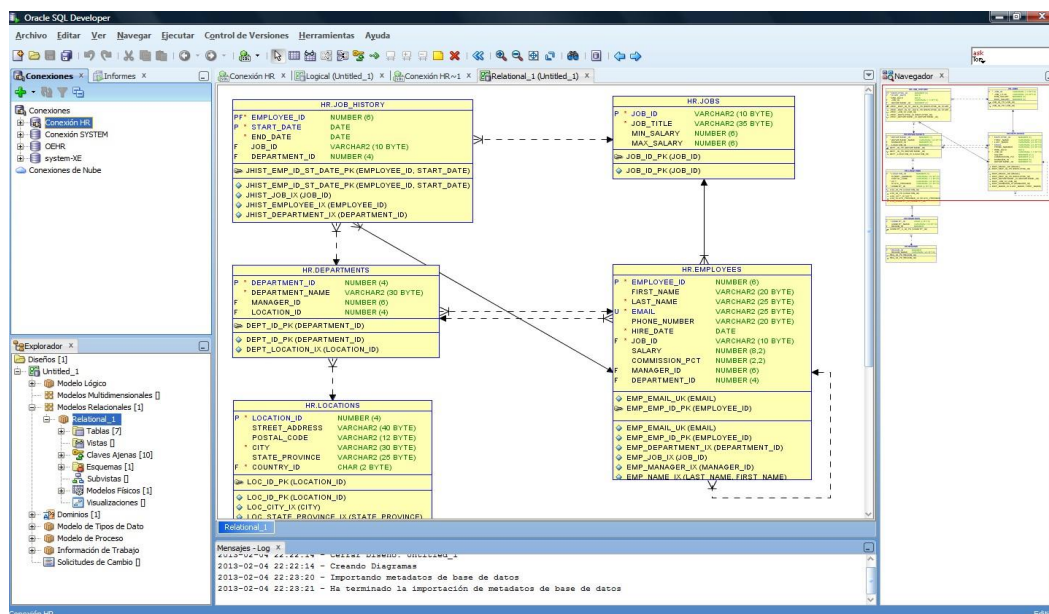


Ilustración 9: Modelo relacional generado con Data Modeler por ingeniería inversa

Se debe activar la vista *Diseños* (en menú *Ver / Data Modeler / Explorador*, también se puede activar *Navegador*). Toda la gestión de ficheros de modelado (abrir, guardar, etc.) se debe hacer a partir del menú *Archivo / Data Modeler / Abrir, Guardar*, etc.



En la ventana de *Explorador* aparecen tanto los modelos relaciones como lógicos que podamos tener cargados. Sobre el modelo relacional, si pulsamos botón derecho tenemos la opción de generar el modelo lógico (opción *Realizar Ingeniería a Modelo Lógico*).

Adicionalmente se pueden imprimir los diagramas generados a través de la opción de menú *Archivo / Data Modeler / Imprimir Diagrama*. Permite formatos: PNG, JPG, SVG y PDF.

Ejercicio

1. Utilizar el *Data Modeler* de SQLDeveloper para generar el correspondiente modelo relacional y entidad-relación de la base de datos HR. Comparar con los resultados obtenidos manualmente con SQL*Plus.

3.6 Algunas diferencias com PostgreSQL

Dado que en el segundo curso los alumnos han trabajado con *pgAdmin* y *PostgreSQL* en la asignatura de *Bases de Datos*, se ha creído conveniente advertir algunas diferencias:

1. Tanto en SQL*Plus como SQLDeveloper se ejecutan todos los comandos del *script* aunque alguno anterior haya dado error. **Si algún comando da error nos sale el mensaje de error, pero intenta ejecutar el siguiente comando.** Por el contrario, en *pgAdmin* si la ejecución de algún comando del *script* da error, el *script* no tiene efecto alguno sobre la base de datos.
2. Tanto en SQL*Plus como SQLDeveloper tenemos eco en pantalla del resultado de todos los comandos incluyendo las *SELECT* (si hay varias *SELECT* se ve el resultado de todas), mientras que en *pgAdmin* sólo se ve el resultado del último comando SQL.
3. Los *scripts* SQL de *PostgreSQL* típicamente empezaban borrando las tablas con *DROP TABLE IF EXISTS Tabla1, Tabla2, etc ... CASCADE*. **Sin embargo con el DROP TABLE de Oracle:**
 1. No hay la opción *IF EXISTS* (recordemos que no es una opción estándar ANSI/ISO), pero nos debería dar igual; pues no la necesitamos, ya que como hemos comentado se ejecutan todos los comandos, por lo que si no existe la tabla a borrar, saldrá simplemente un mensaje de error, pero se ejecutarán el resto de comandos.
 2. No se pueden borrar varias tablas con el mismo *DROP TABLE*, hay que hacer un *DROP TABLE* separado para cada una.
 3. La opción *CASCADE* en Oracle tiene la sintaxis *CASCADE CONSTRAINTS*.

Por todas estas razones habría que cambiar el *DROP TABLE* de PostgreSQL:

```
DROP TABLE IF EXISTS Tabla1, Tabla2, etc ... CASCADE;
```

por:

```
DROP TABLE Tabla1 CASCADE CONSTRAINTS;  
DROP TABLE Tabla2 CASCADE CONSTRAINTS;  
etc ...
```

3.7 Oracle Application Express (APEX)

APEX es una herramienta de creación de formularios e informes web para Oracle. **En Oracle 11g, por defecto, viene instalada una versión 4 de APEX, que ha de ser actualizada a una versión 5 (versión 5.1.4) mediante una serie de pasos adicionales.** En la máquina virtual utilizada en la asignatura **ya está actualizado.**

Por otro lado se proporciona un **manual de instalación y actualización de las herramientas utilizadas en la asignatura**, donde se indican y detallan dichos pasos previos.



Así pues, los pasos y captura de pantalla que se muestran a continuación se realizan sobre dicha versión 5 ya actualizada. **OJO, aunque existen nuevas versiones de APEX están pensadas para su uso con Oracle Express 18c o superior, versión que NO se utilizará en esta asignatura. Por lo tanto la versión de APEX a utilizar debe ser siempre la 5.1.4. Es muy importante trabajar todos con la misma versión.**

Partiendo de dichos pasos, y suponiendo que nuestro administrador de APEX se ha definido como *Admin* con *password Abd2019**, vamos a crear un *workspace* de ejemplo.

En primer lugar nos conectamos en un navegador en la siguiente URL para la administración de APEX:
http://localhost:8080/apex/apex_admin

Se debe mostrar una pantalla de bienvenida similar a:

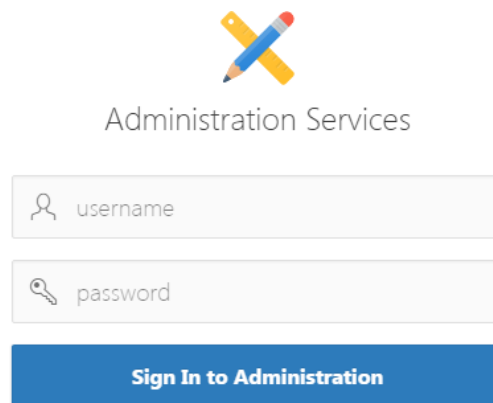


Ilustración 10: Validación de administrador de APEX

Introducimos el *login* (*Admin*) y *password* (Ej: *Abd2019**). Se nos da la bienvenida a la consola web de administración de APEX con una barra de herramientas similar a la siguiente:

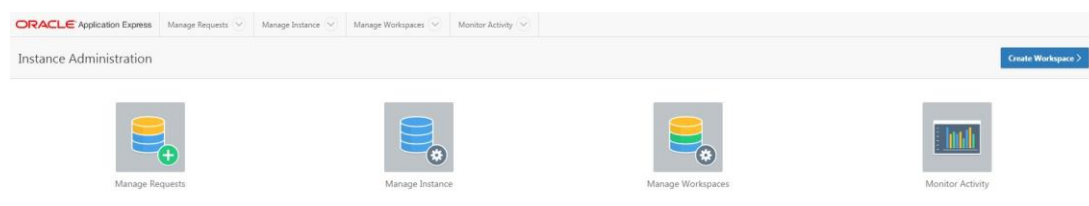


Ilustración 11: Consola de administración web de APEX

Seleccionamos el botón *Manage Workspaces*, que nos lleva a una pantalla con el siguiente menú en la parte izquierda.



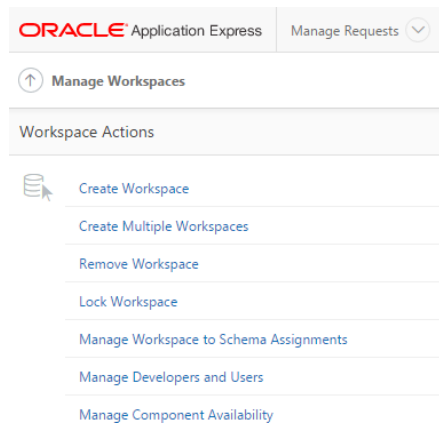


Ilustración 12: Menú de gestión de workspaces

Seleccionamos la opción *Create Workspace* que nos va a guiar mediante una serie de pasos a la creación de nuestro primer *workspace* de prueba. Introducimos el nombre (por ejemplo, *Prueba*) y pulsamos en *Next* (el resto de campos de momento no se rellenarán).

Ilustración 13: Primer paso en la creación de un workspace

A continuación elegimos Yes a la hora de reutilizar un esquema (*schema*) existente (*Re-using existing schema?*), por ejemplo HR, seleccionamos el esquema (no hace falta en este caso introducir *password*) y pulsamos en *Next*.



The screenshot shows the 'Create Workspace' wizard at the 'Identify Schema' step. A progress bar at the top has three steps: the first is completed (green dot), the second is active (blue dot), and the third is pending (grey dot). Below the progress bar, instructions state: 'Select whether or not the schema already exists. If the schema exists, select the schema from the list. If the schema does not exist, enter a name and password and choose the size of the associated tablespace to be created.'

The form contains the following fields:

- 'Re-use existing schema?' with a dropdown menu set to 'No'.
- 'Schema Name' with a text input field.
- 'Schema Password' with a text input field.
- 'Space Quota (MB)' with a dropdown menu set to '100'.

At the bottom, there are three buttons: a back arrow, a 'Cancel' button, and a 'Next >' button.

Ilustración 14: Segundo paso en la creación de un workspace

Creamos la cuenta de administrador del *workspace* (ojo, es un **administrador del workspace**, **NO de APEX**, son dos **roles completamente diferentes**), y se le da un *password* (aunque en el primer acceso nos obligará a cambiarlo). Rellenamos los datos (pueden ser ficticios) y presionamos en *Next*.

The screenshot shows the 'Create Workspace' wizard at the 'Identify Administrator' step. The progress bar now has two completed steps (green dots) and one active step (blue dot). The form contains the following fields:

- 'Administrator Username' with a text input field containing 'ADMIN'.
- 'Administrator Password' with a text input field.
- 'First Name' with a text input field.
- 'Last Name' with a text input field.
- 'Email' with a text input field containing 'admin@vubues'.

At the bottom, there are three buttons: a back arrow, a 'Cancel' button, and a 'Next >' button.

Ilustración 15: Tercer paso en la creación de un workspace

Finalmente confirmamos los datos y la creación del *workspace*.



Create Workspace

Confirm Request

You have requested to provision a new Workspace.

Workspace Information:

Name	Borrar
Workspace ID	System Assigned
Description	...

Administrator Information:

User Name	ADMIN
E-mail	admin@ubu.es

Schema Information:

Reuse Existing Schema	Yes
Schema Name	HR

Ilustración 16: Cuarto paso en la creación de un workspace

Se confirma la creación del *workspace* y se nos pregunta si queremos editarlo o seguir en la administración, pulsamos en *Done*.

Workspace Created

Workspace Borrar provisioned with administrator ADMIN.

Ilustración 17: Confirmación de creación del workspace

Cerramos la sesión como **administradores de APEX** (en la esquina superior derecha, donde aparece un icono de la silueta de un usuario, desplegamos el menú y seleccionamos la opción *Sign Out*).

Probamos ahora a conectarnos ahora como **administradores del nuevo workspace** que acabamos de crear. Para ello accedemos a la URL de acceso como *administrador de workspace*: <http://localhost:8080/apex/> mostrándose una ventana de validación. En este caso se pregunta adicionalmente por el *workspace* al que queremos acceder.



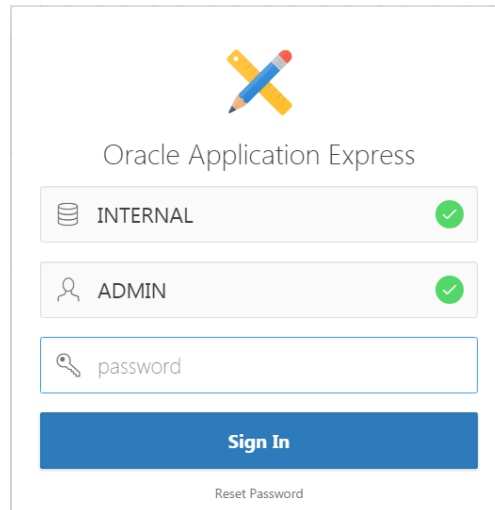


Ilustración 18: Ventana de validación para acceder a un workspace

Introducimos el nombre del *workspace* que acabamos de crear (cambiamos `INTERNAL` por `Prueba`) y los datos de acceso del usuario *admin* del *workspace*. En el primer acceso nos pedirá que cambiemos el *password* inicialmente fijado.

Se nos mostrará una pantalla con las siguientes opciones dentro del *workspace*:

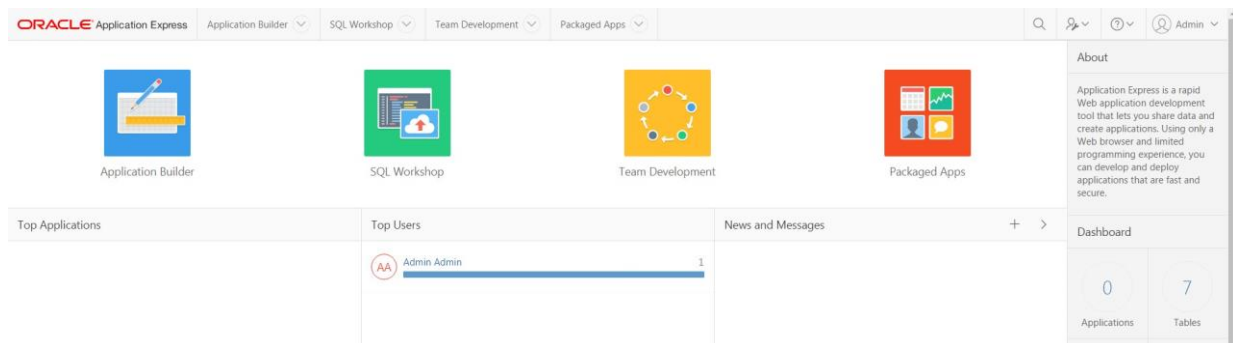


Ilustración 19: Barra de herramientas en un workspace

Seleccionamos el botón *App Builder* (o *Application Builder*). Dicha herramienta nos permitirá **generar aplicaciones web con nuestros formularios e informes**.

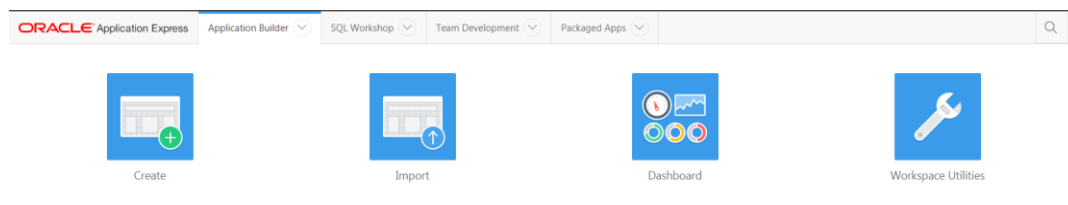


Ilustración 20: Opciones de Application Builder

Finalmente, vamos a ver cómo se exporta un *workspace*. Para ello seleccionamos *Workspace Utilities* a la derecha (botón con la llave inglesa).



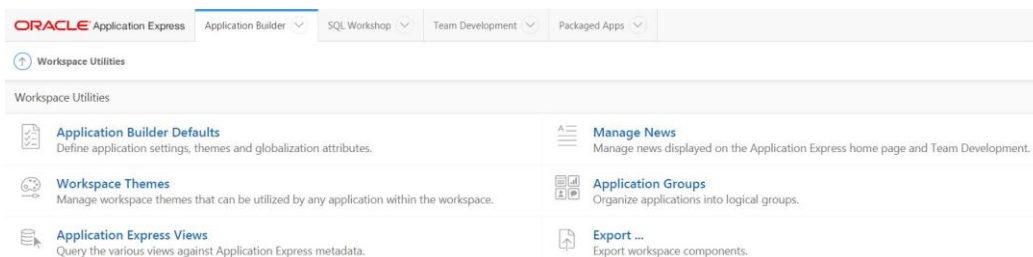


Ilustración 21: Opciones de Workspace Utilities

Dentro de las opciones seleccionamos *Export* y elegimos *Workspace*:

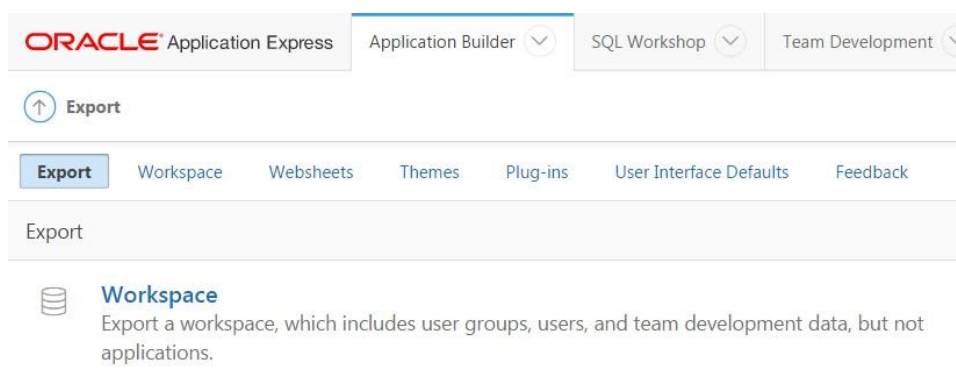


Ilustración 22: Exportación de un workspace

Se nos preguntará por una serie de parámetros adicionales para el fichero de exportación y pulsamos el botón *Export*.

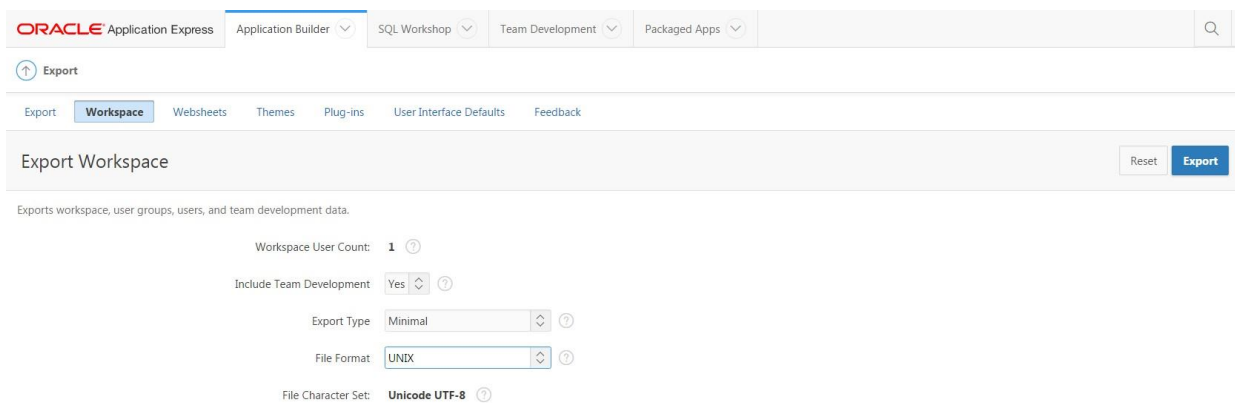


Ilustración 23: Ajustes para la exportación de un workspace

Confirmando la acción en el botón *Export*, se genera un script SQL que permite la importación en otras máquinas que APEX de dicho *workspace*.

Para cerrar sesión de administrador de un *workspace*, al igual que antes, en la esquina superior derecha, seleccionamos sobre el botón del usuario, la opción *Sign Out*.

Finalmente, podemos acceder de nuevo como **administradores de APEX** en:

http://localhost:8080/apex/apex_admin

En la barra de herramientas, podemos seleccionar *Manage Workspaces/Remove Workspace* si queremos borrar el *workspace* creado previamente



Ejercicio

1. Realizar los pasos previos para llegar a obtener el `script.sql` resultante de la exportación del *workspace*.

4. Resumen

En este guión de prácticas se ha realizado una **primera toma de contacto** y planteado las actividades básicas iniciales para empezar a trabajar con Oracle desde el punto de vista de un desarrollador de aplicaciones. Para ello se han presentado tres herramientas básicas que se utilizarán en la asignatura: SQL*Plus, SQLDeveloper y APEX.

5. Glosario

APEX: Application Express.

SGBD: Sistema Gestor de Bases de Datos.

Workspace: espacio de trabajo en APEX para crear aplicaciones.

6. Recursos

Para una consulta adicional de información, se recomiendan los siguientes enlaces:

[Oracle, 2013] Oracle (2013) Oracle SQL*Plus User's Guide and Reference . Disponible en https://docs.oracle.com/cd/E11882_01/server.112/e16604/qstart.htm#SQPUG002

[Oracle, 2019] Oracle (2019) Oracle SQLDeveloper Documentation. Disponible en <http://www.oracle.com/technetwork/developer-tools/sql-developer/documentation/index.html>

[Oracle, 2019] Oracle (2019) Oracle Application Express. Disponible en <https://apex.oracle.com/es/>

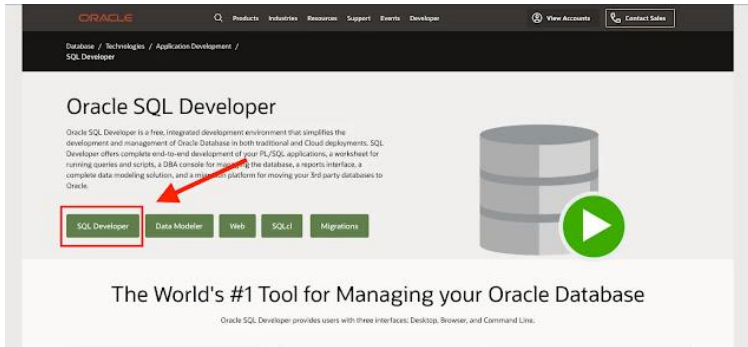


ANEXO

INSTRUCCIONES DE INSTALACIÓN SQL Developer EN LA MÁQUINA VIRTUAL DE LA ASIGNATURA

IMPORTANTE: Estas instrucciones han sido redactadas para ser ejecutadas en la máquina virtual de la asignatura, y en base a la configuración inicial de esta. No se asegura que estos pasos sean suficientes para una instalación en otra máquina. Si deseas instalarlo en tu máquina, deberás consultar los requisitos de instalación en la página del proveedor.

1. Acceder a la web de descarga
<https://www.oracle.com/database/technologies/appdev/sqldeveloper-landing.html>
2. Click en el botón SQL Developer



3. Descargar el paquete para Linux RPM (os pedirá crear una cuenta de Oracle o usar una que tengáis)
4. En un terminal, acceder a la carpeta donde se haya descargado y ejecutar:

```
sudo alien --scripts -d sqldeveloper-21.2.1-204.1703.noarch.rpm
```

5. Esto genera el archive .deb. Para ejecutar ese archivo ejecutar:

```
sudo dpkg --install sqldeveloper_21.2.1-205.1703_all.deb
```



Licencia

Autores: Jesús Maudes & Raúl Marticorena

Área de Lenguajes y Sistemas Informáticos

Departamento de Ingeniería Civil

Escuela Politécnica Superior

UNIVERSIDAD DE BURGOS

2020



Este obra está bajo una licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Unported. No se permite un uso comercial de esta obra ni de las posibles obras derivadas, la distribución de las cuales se debe hacer con una licencia igual a la que regula esta obra original

Licencia disponible en <http://creativecommons.org/licenses/by-nc-sa/4.0/>

