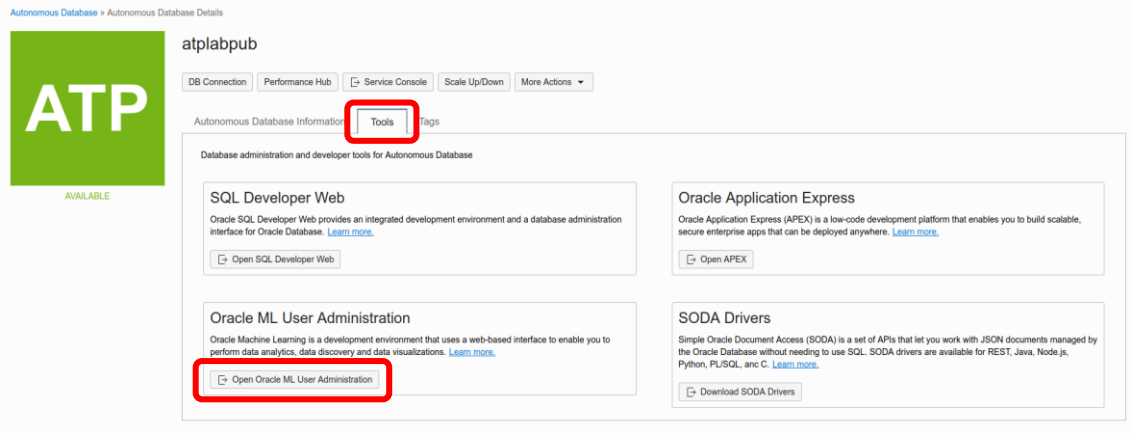
HOL5 - ATP tooling

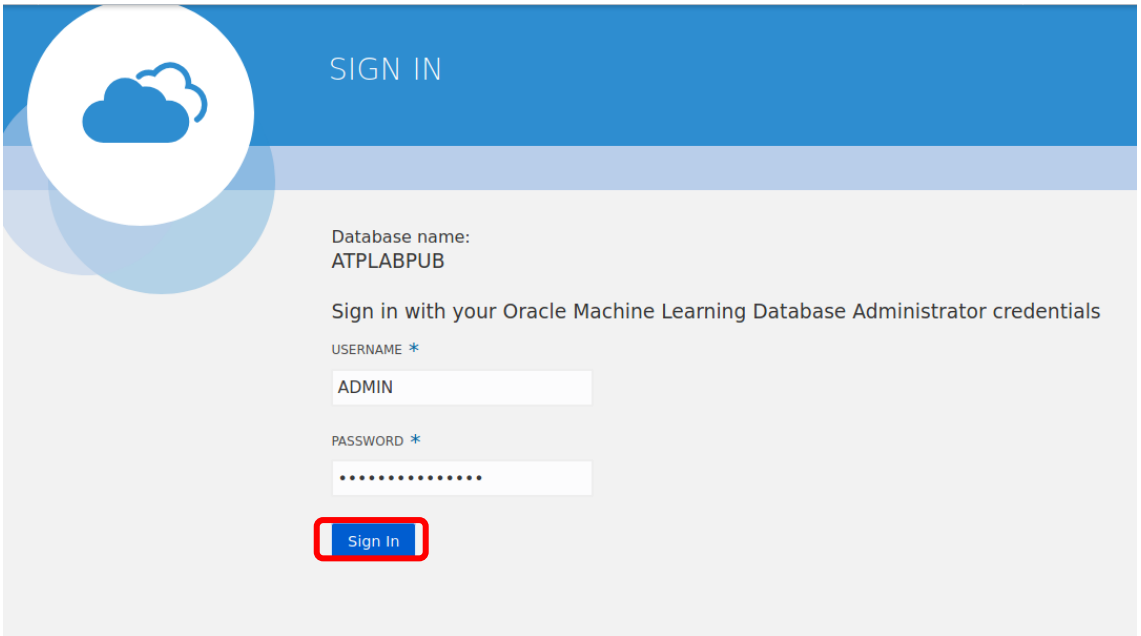
INDICE

Ejercicio 1: Crear usuario de Machine Learning

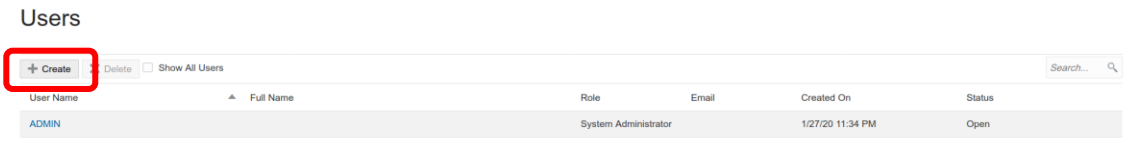
En este ejercicio se explica crear un usuario de Oracle Machine Learning. Este usuario lo utilizaremos en ejercicios posteriores para crear un Notebook y consultar datos del esquema HR.
En la pantalla principal del ATP, elegir la pestaña "**Tools**", y luego cliquar el botón "**Open Oracle ML User Administration**"



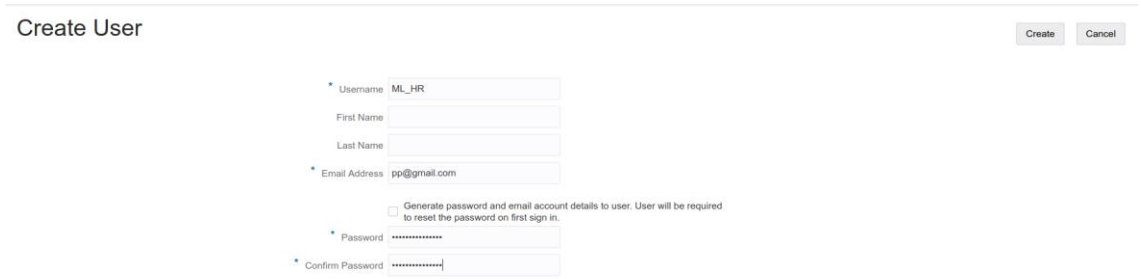
En la pantalla de login, entrar las credenciales de ADMIN:
Usuario: ADMIN
Contraseña: Autonomous#2020



En la pantalla siguiente, vemos un listado de usuarios de OML, solo ADMIN de momento.

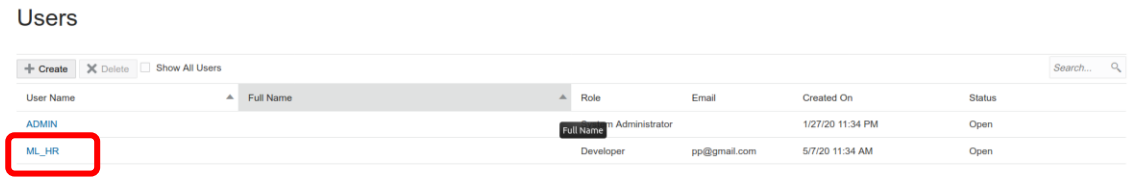


Cliquar el botón "**Create**".



En la pantalla de creación de usuario, rellenar la información y pulsar el botón "**Create**".
Usuario: ML_HR
Password: Autonomous#2020
Email: cualquier valor con formato email.
Importante: de-chequear "*Generate password and email account details to user*", para poder teclear la contraseña.

Una vez creado, el nuevo usuario aparece en la lista de usuarios de OML:

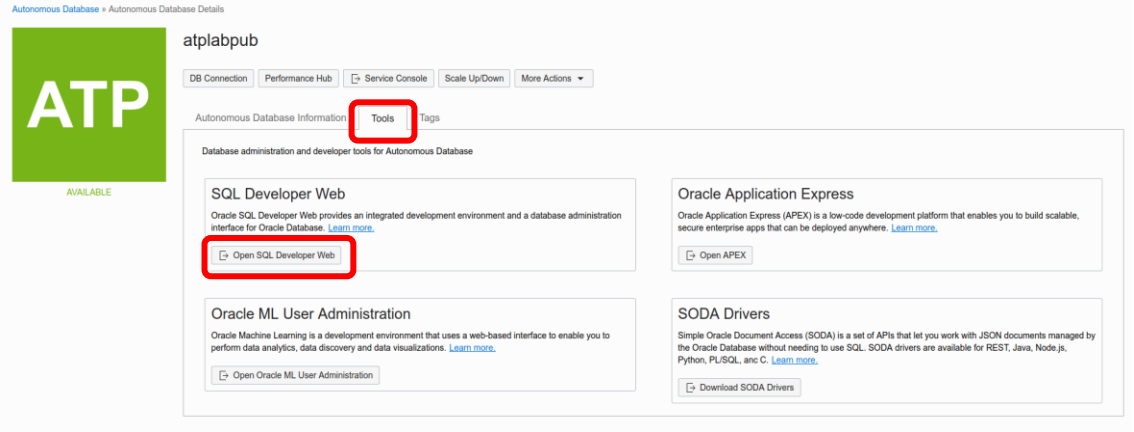


Ejercicio 2: Utilizar Sql*Developer Web

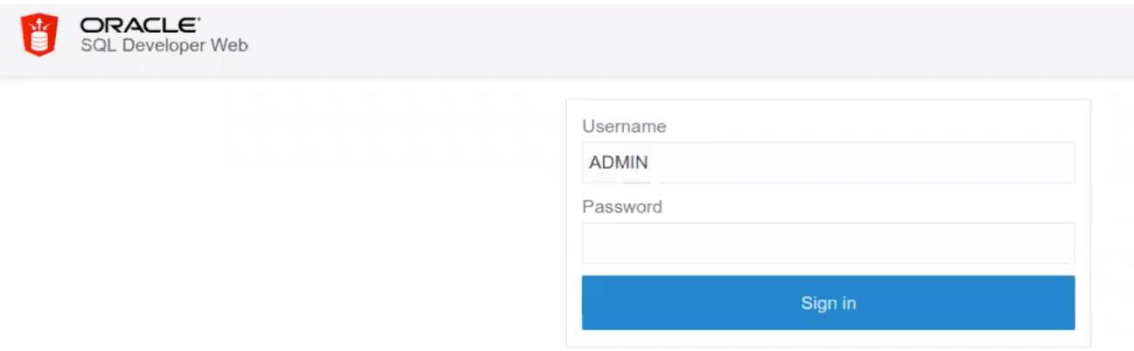
En este ejercicio, vamos a utilizar Sql*Developer Web para:

- Habilitar ORDS para el esquema HR
- Como HR, ejecutar algunas consultas SQL
- Habilitar una política de Data Redaction sobre un campo de la tabla “employees”
- Dar privilegios de consulta al usuario ML_HR sobre la tabla “employees”
- Habilitar ORDS sobre la tabla “employees”

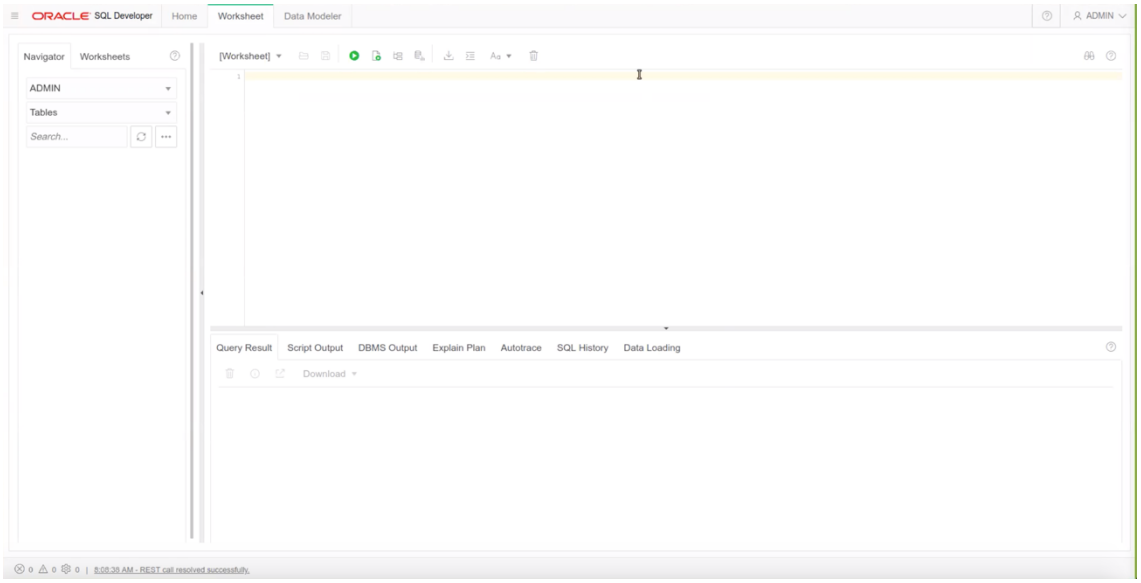
Desde la pantalla principal del ATP, en la pestaña “**Tools**”, cliquar el botón “**Open Sql Developer Web**”.



Esto nos lleva a la pantalla de login: conectarse como usuario ADMIN, password “Autonomous#2020”.

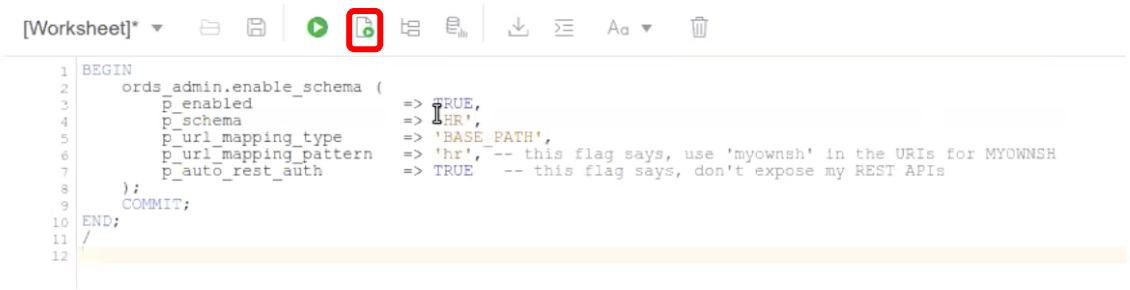


Una vez hecho esto, accedemos a la pantalla de SQL Developer Web, Que nos dará acceso para interactuar con la base de datos mediante SQL.



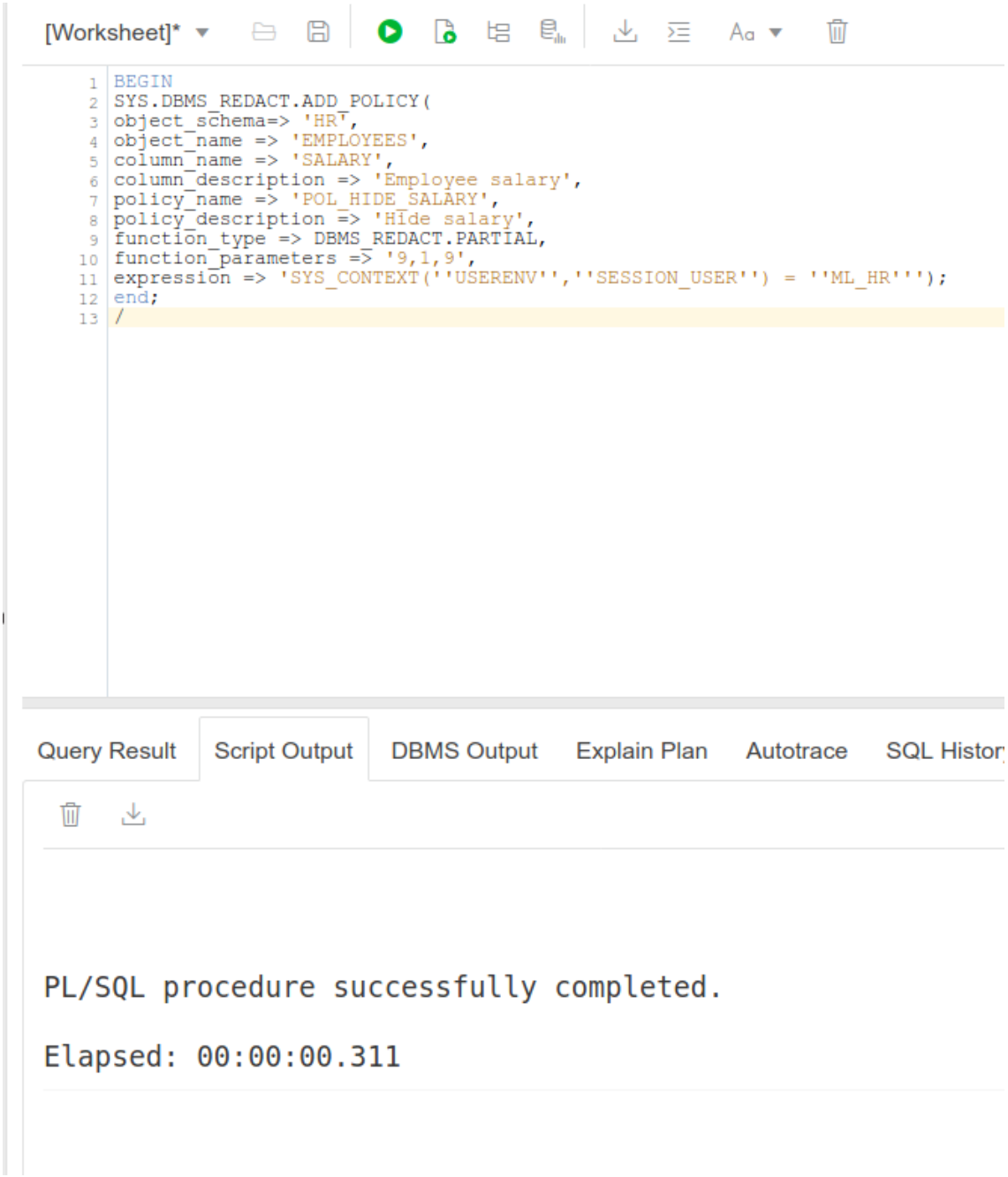
A continuación, ejecutamos el siguiente código, que habilita el usuario HR en ORDS y lo habilita para poder acceder con SQL Developer Web:

```
BEGIN
    ords_admin.enable_schema (
        p_enabled               => TRUE,
        p_schema                => 'HR',
        p_url_mapping_type      => 'BASE_PATH',
        p_url_mapping_pattern   => 'hr',
        p_auto_rest_auth        => TRUE    );
    COMMIT;
END;
/
```



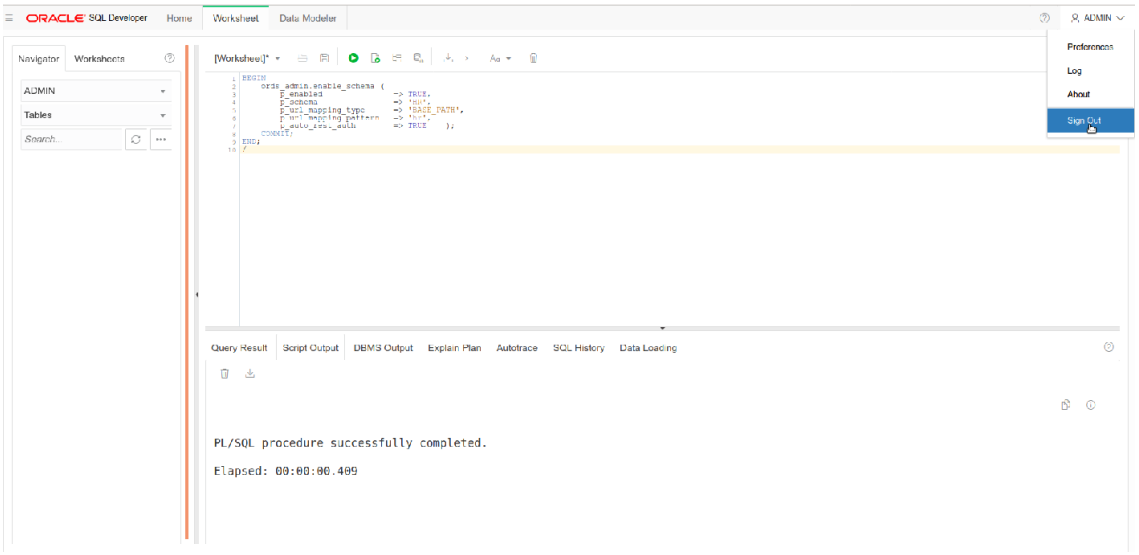
Luego, para preparar un ejercicio posterior, vamos a habilitar Data Redaction sobre la tabla "HR.employees", para impedir que el usuario ML_HR pueda ver el contenido de la columna "salary". Ejecutamos el código siguiente:

```
BEGIN
SYS.DBMS_REDACT.ADD_POLICY(
object_schema=> 'HR',
object_name => 'EMPLOYEES',
column_name => 'SALARY',
column_description => 'Employee salary',
policy_name => 'POL_HIDE_SALARY',
policy_description => 'Hide salary',
function_type => DBMS_REDACT.PARTIAL,
function_parameters => '9,1,9',
expression => 'SYS_CONTEXT(''USERENV'',''SESSION_USER'') = ''ML_HR''');
end;
/
```



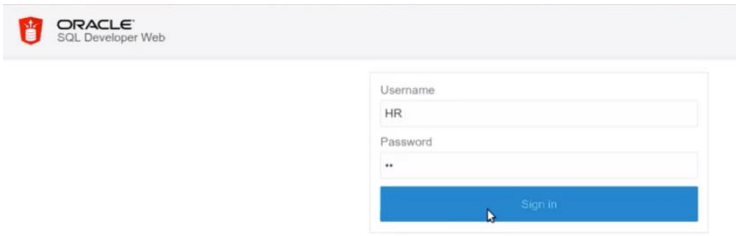
Esta política impedirá que el usuario ML_HR pueda ver el contenido del campo “salary” en la tabla “HR.employees”.

Una vez hecho esto, podemos acceder a SQL Developer Web mediante la URL anterior, pero cambiando el usuario admin por HR.
Primero nos desconectamos del Sql*Developer Web, y cerramos la pestaña del navegador.

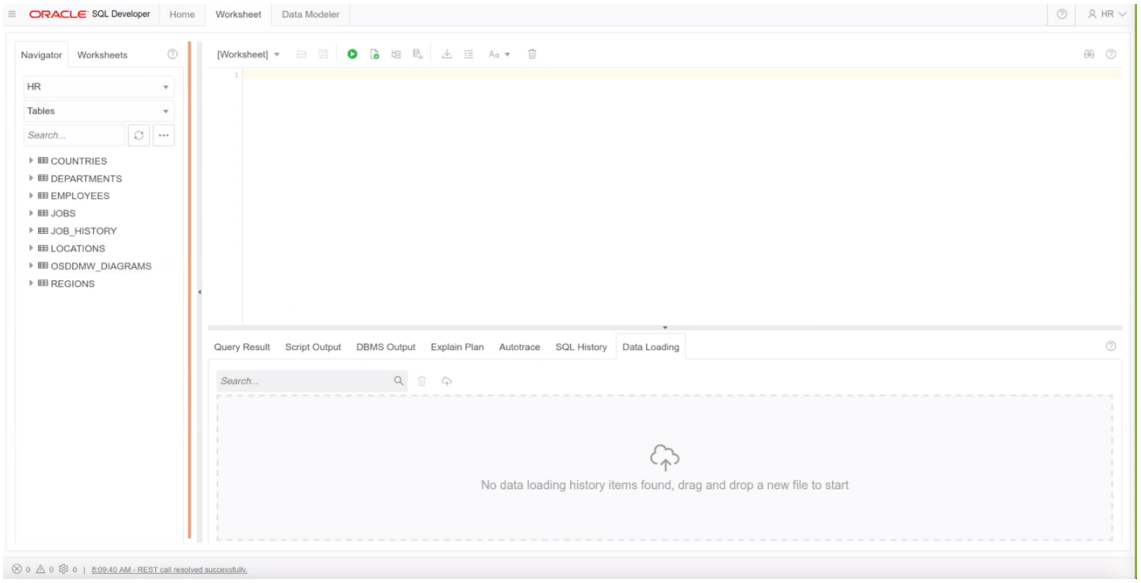


Luego desde la pagina principal del ATP, pestaña "**Tools**", volvemos a cliquar "**Open Sql Developer Web**", para volver a la pantalla de login. En la URL, cambiamos "admin" por "hr":

Volverá a a aparecer la consola de login, volvemos a introducir el nombre de usuario y contraseña. En este caso el usuario HR/hr o hr/hr (**contraseña siempre en minúsculas**).

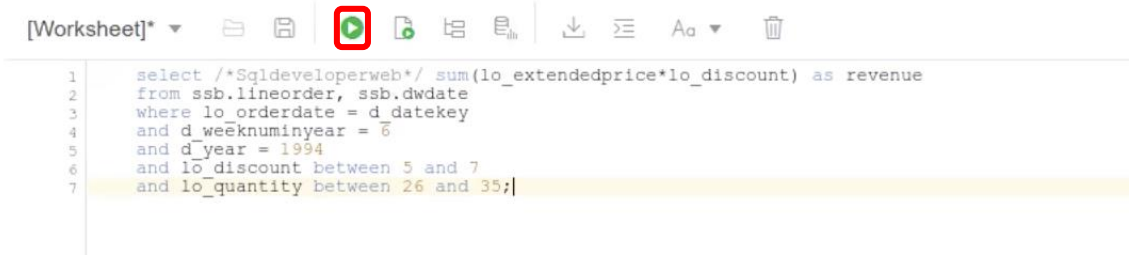


En la siguiente pantalla podemos ver la misma consola de SQL Developer Web, pero en este caso a la izquierda podemos ver las tablas del esquema HR:

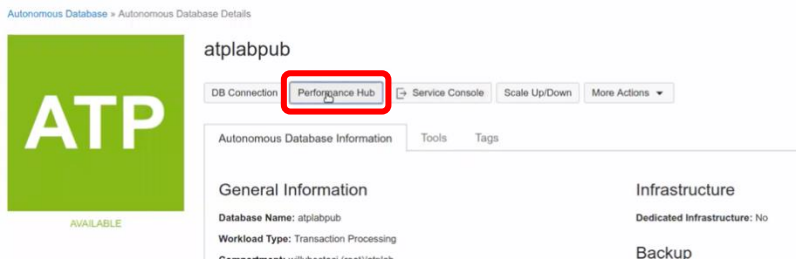


Ejecutamos el siguiente código, como se muestra en la imagen. Es una query sobre el esquema SSB, accesible a cualquier usuario:

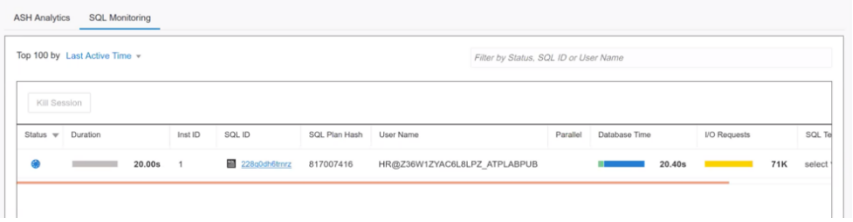
```
select /*Sqldeveloperweb*/ sum(lo_extendedprice*lo_discount) as revenue
from ssb.lineorder, ssb.dwdate
where lo_orderdate = d_datekey
and d_weeknuminyear = 6
and d_year = 1994
and lo_discount between 5 and 7
and lo_quantity between 26 and 35;
```



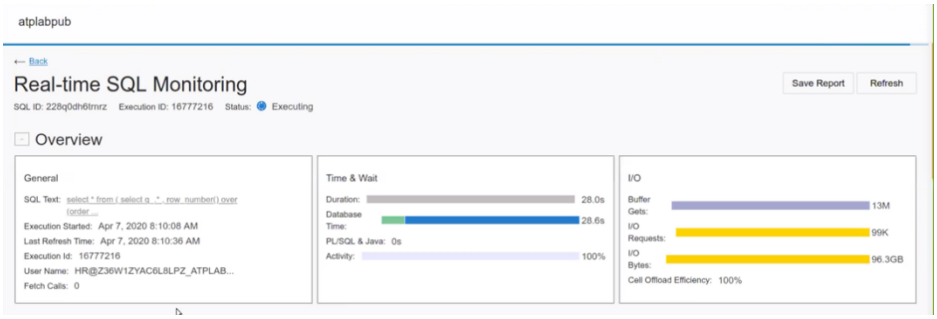
Se puede consultar la ejecución de la query desde la sección Performance HUB, en la pantalla principal del ATP:



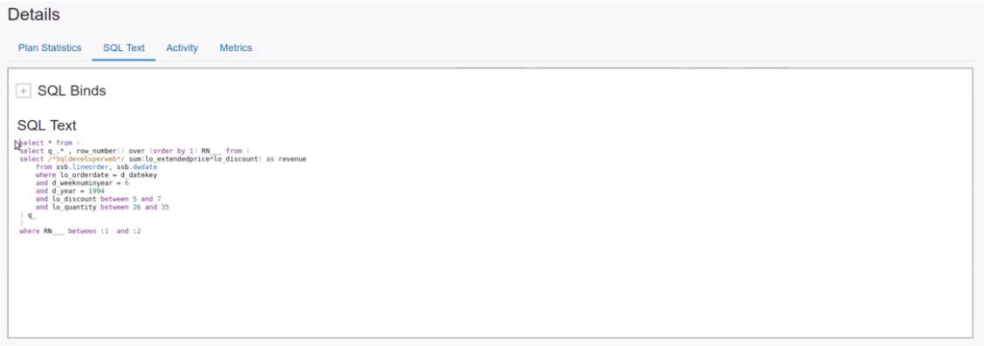
Una vez aquí, se puede ver la query ejecutándose:



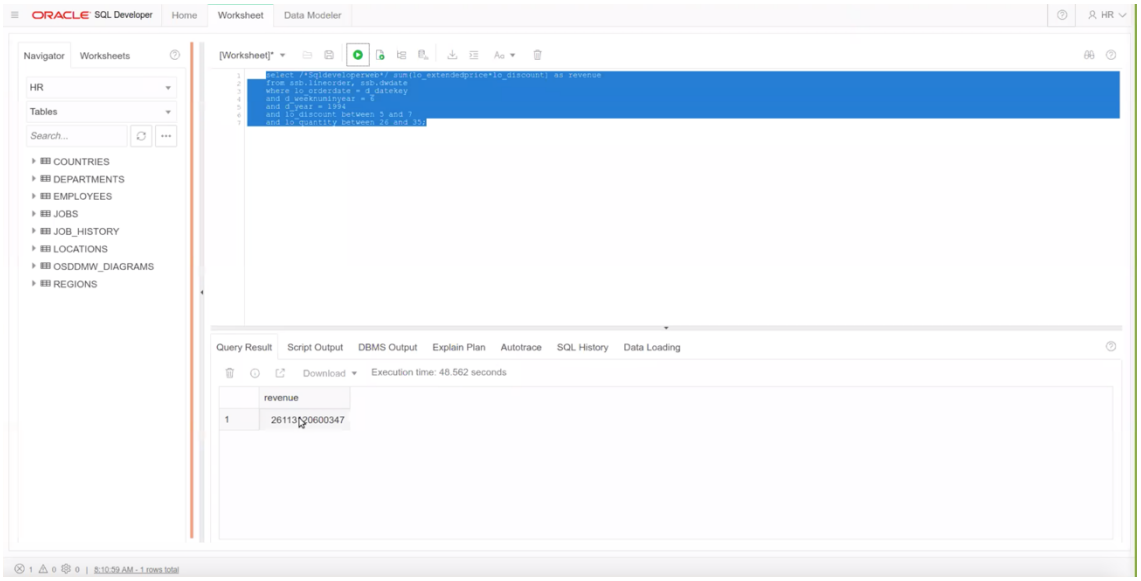
Se puede seleccionar el SQL ID de la query, y acceder a sus detalles



Más abajo, se puede ver el código de la query, en la pestaña SQL Text:

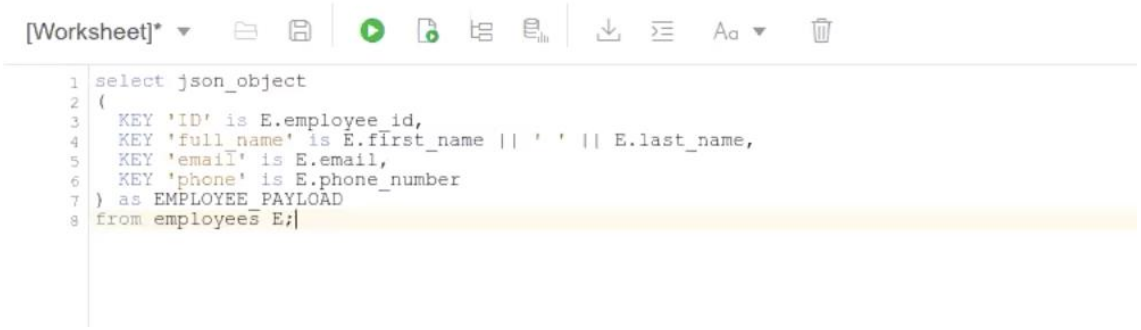


Volviendo a SQL Developer Web, cuando la query haya terminado, podemos ver el resultado de la consulta:

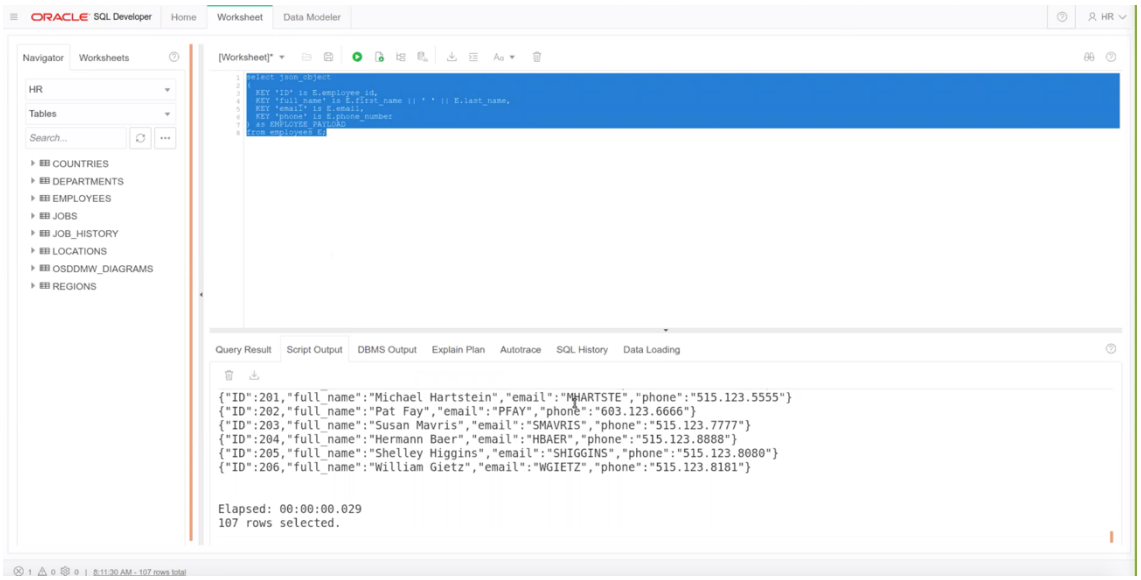


A continuación, ejecutamos una segunda consulta, en este caso la consulta devuelve un objeto JSON a partir de los datos de la consulta SQL:

```
select json_object
(
  KEY 'ID' is E.employee_id,
  KEY 'full_name' is E.first_name || ' ' || E.last_name,
  KEY 'email' is E.email,
  KEY 'phone' is E.phone_number
) as EMPLOYEE_PAYLOAD
from employees E;
```

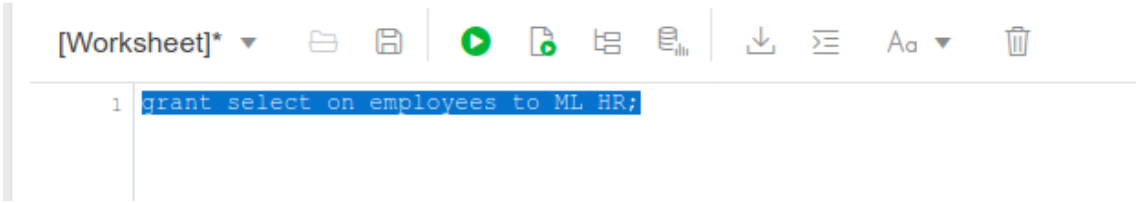


Podemos ver el resultado en formato JSON:



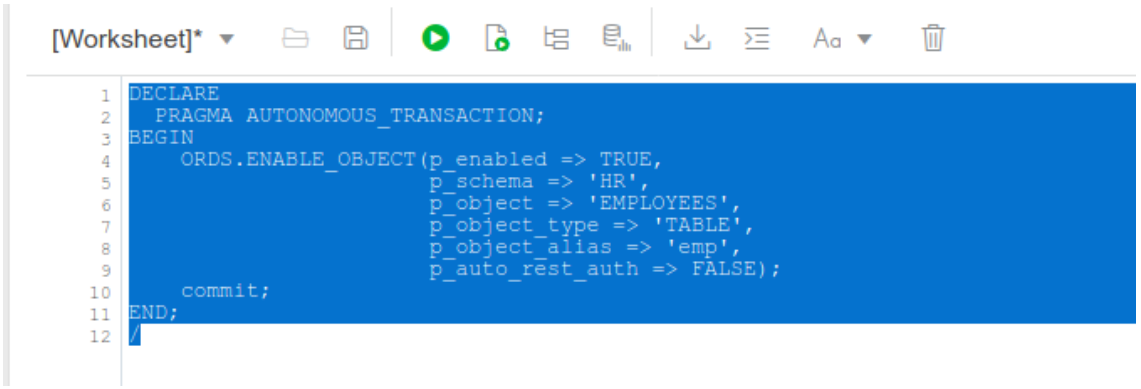
Para preparar los ejercicios siguientes, otorgamos privilegios al usuario ML_HR sobre la tabla "employees":

```
grant select on employees to ML_HR;
```

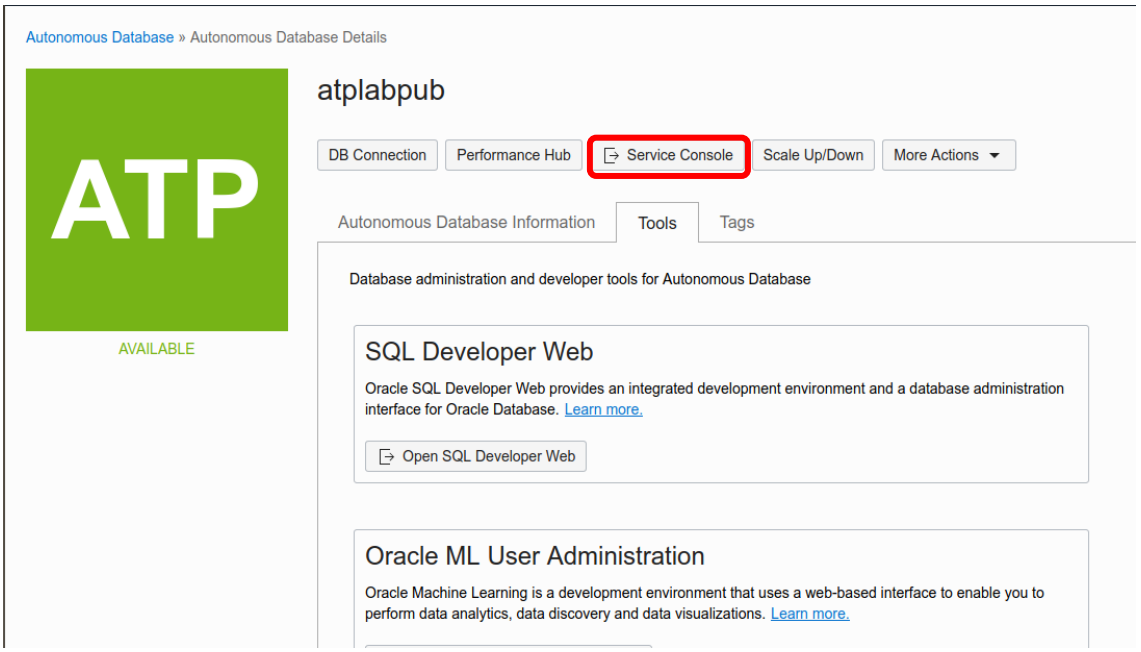


Y habilitamos ORDS sobre la tabla "employees", para permitir el acceso por REST a sus datos:

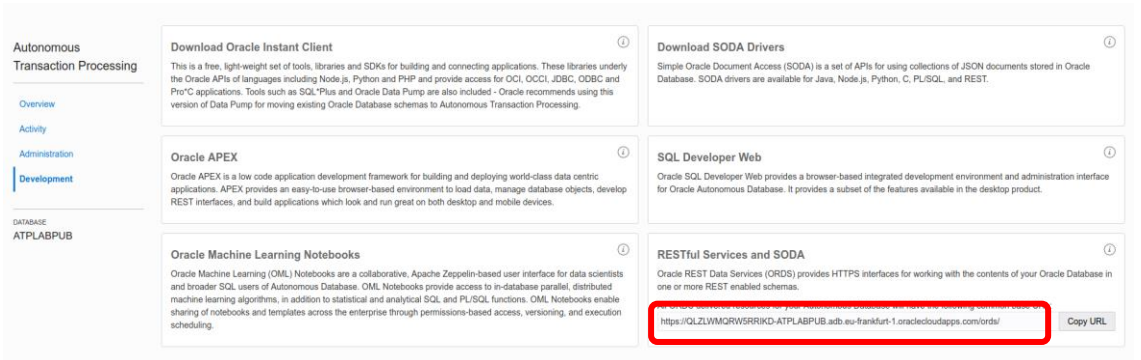
```
DECLARE
  PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
    ORDS.ENABLE_OBJECT(p_enabled => TRUE,
                       p_schema => 'HR',
                       p_object => 'EMPLOYEES',
                       p_object_type => 'TABLE',
                       p_object_alias => 'emp',
                       p_auto_rest_auth => FALSE);
    commit;
END;
/
```



Ahora podremos consultar la tabla “employees” mediante REST API. Para ello recuperamos el REST Endpoint desde la pagina principal del ATP. Hacer click en el botón “Service Console”:

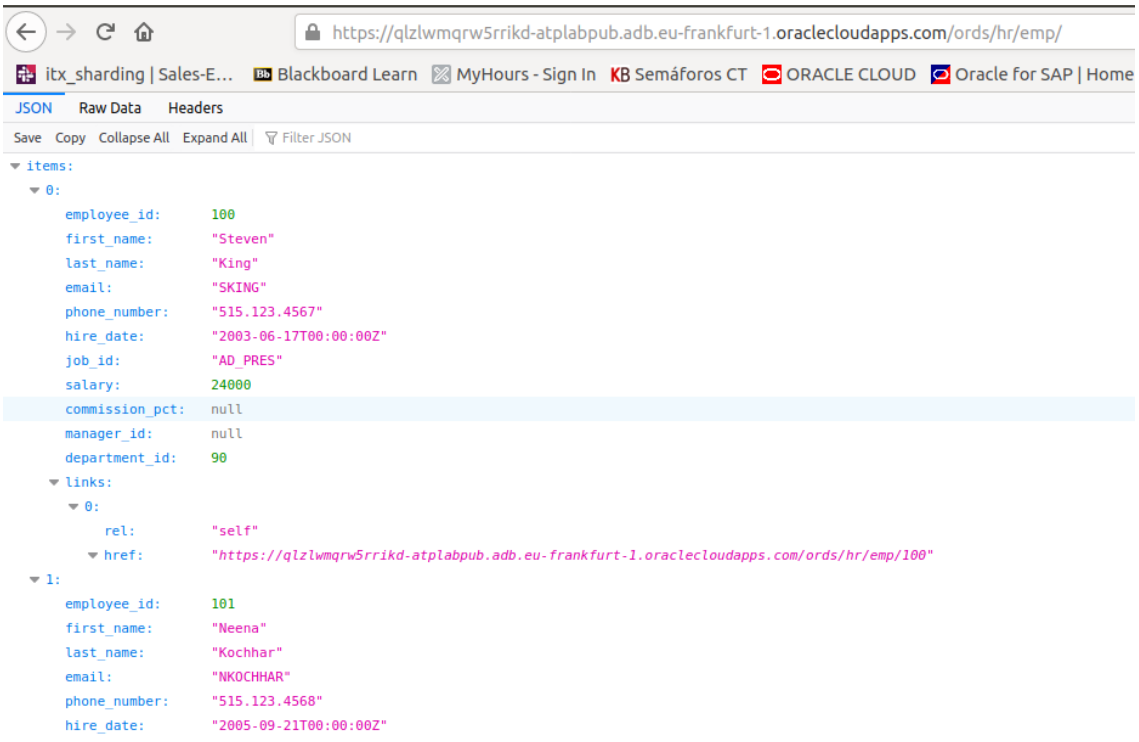


En la pantalla siguiente, hacer click en “Development”. En el apartado “RESTful Services and SODA”, vemos nuestro REST Endpoint:

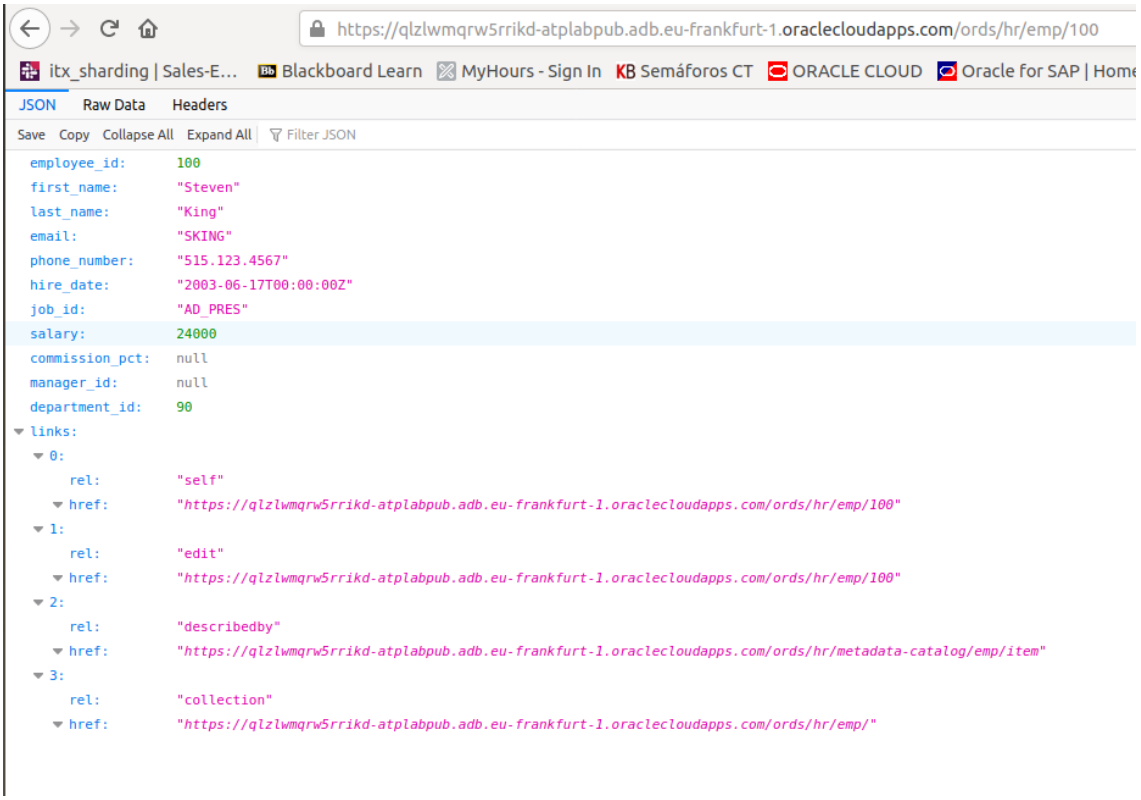


Para consultar nuestra tabla, a la URL de REST Endpoint le añadimos “/hr/emp/”, por ejemplo: https://QLZLWMQRW5RRIKD-ATPLABPUB.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/

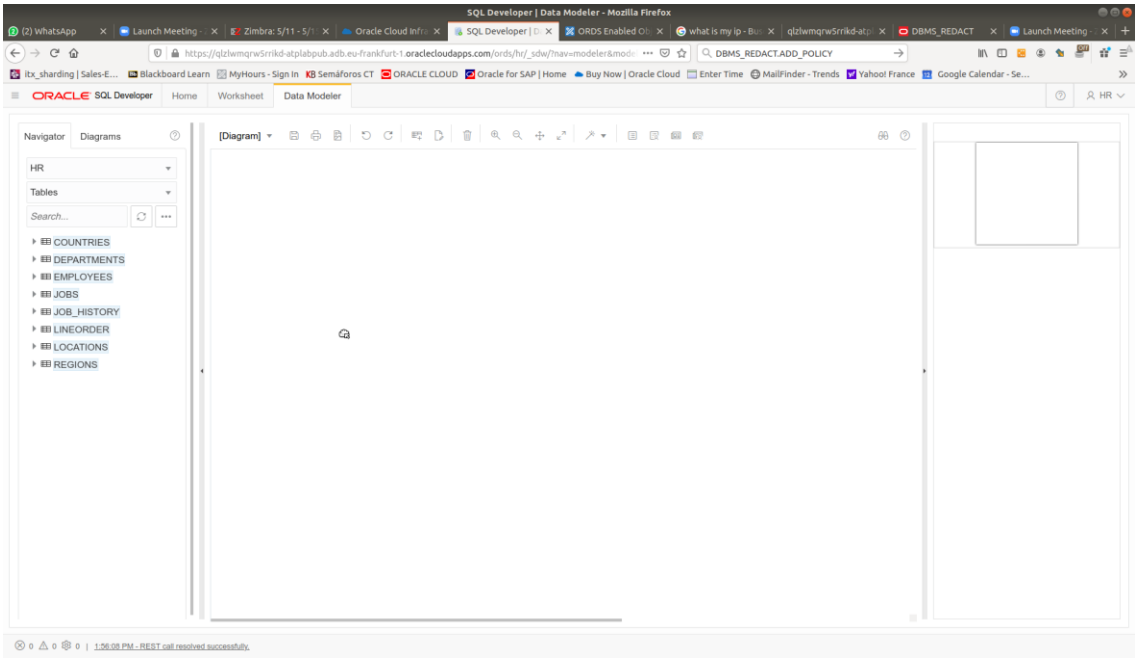
Si pegamos esta URL en un navegador Web, vemos los datos de la tabla “employees”:



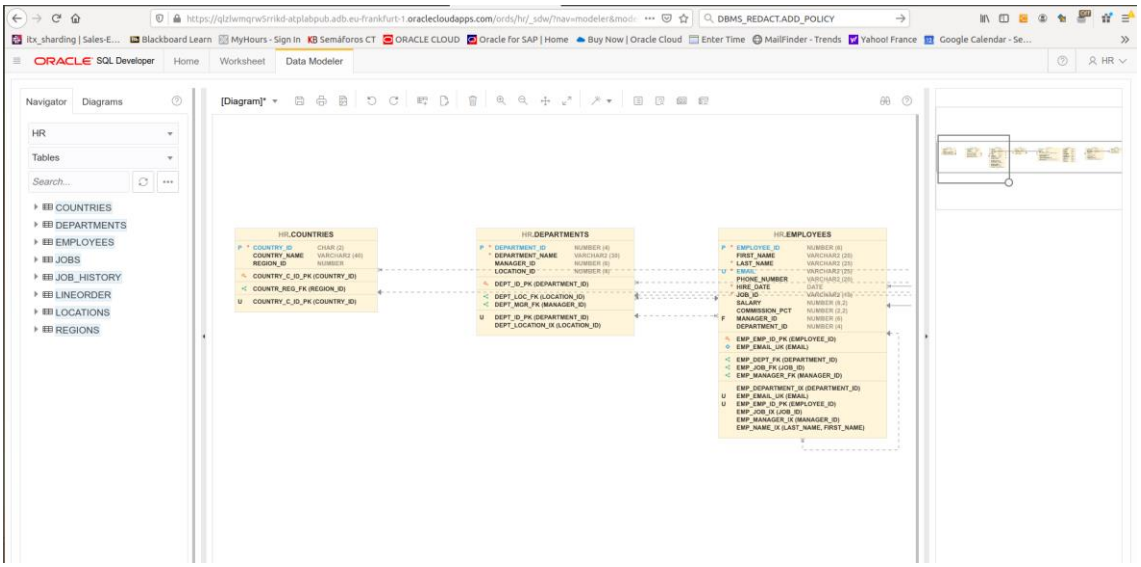
Si queremos ver únicamente el employee ID=100, completamos la URL con “100”:



Finalmente, desde el Sql*Developer Web, hacemos click en la pestaña Data Modeler para visualizar el modelo de datos del esquema HR. Arrastramos todas las tablas a la parte central de la pantalla:



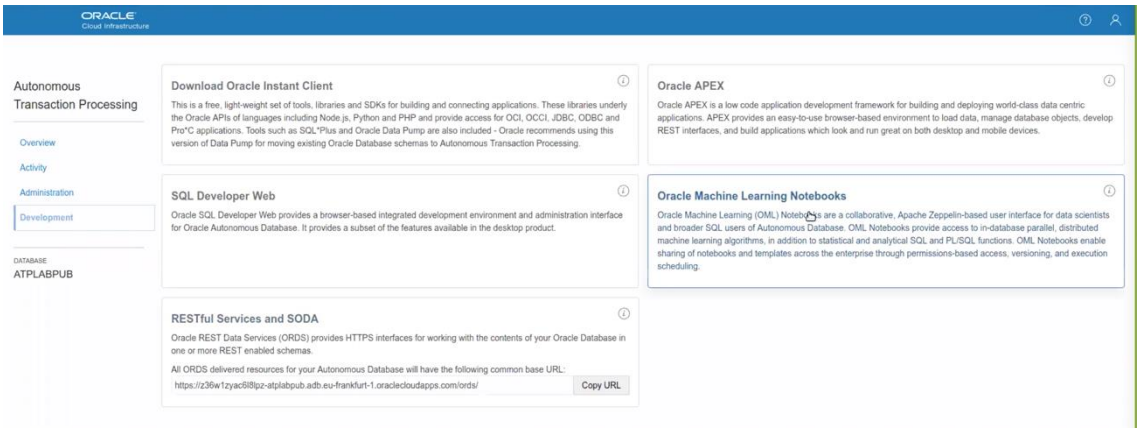
Y visualizamos nuestro modelo relacional:



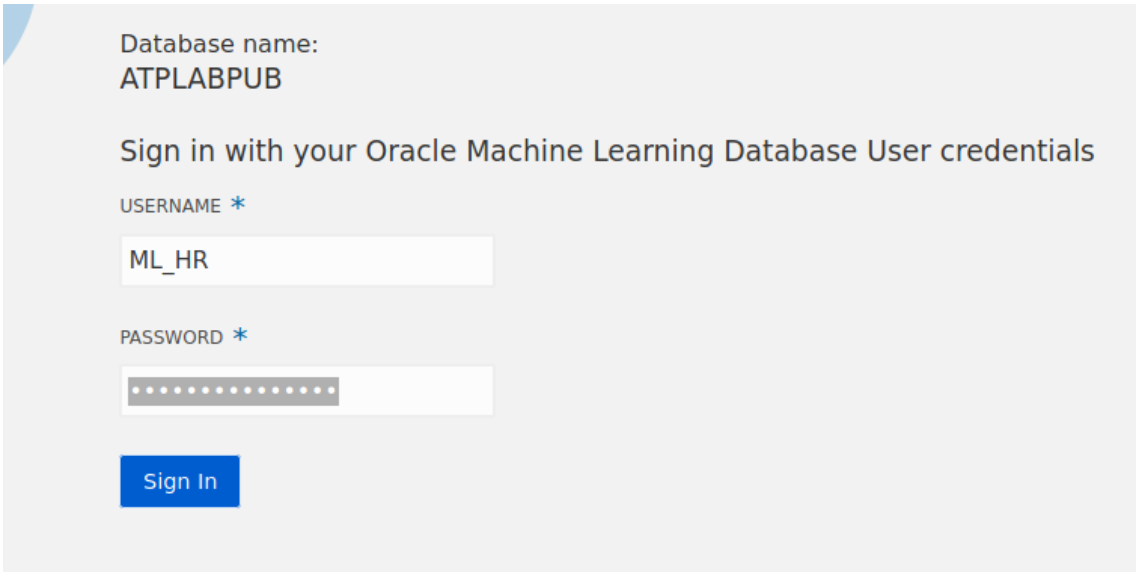
Ejercicio 3: Crear un notebook en Oracle Machine Learning

En este ejercicio vamos a conectarnos a OML con el usuario ML_HR que hemos creado anteriormente. Desde la pantalla principal del ATP. Pulse en el botón "**Service Console**".

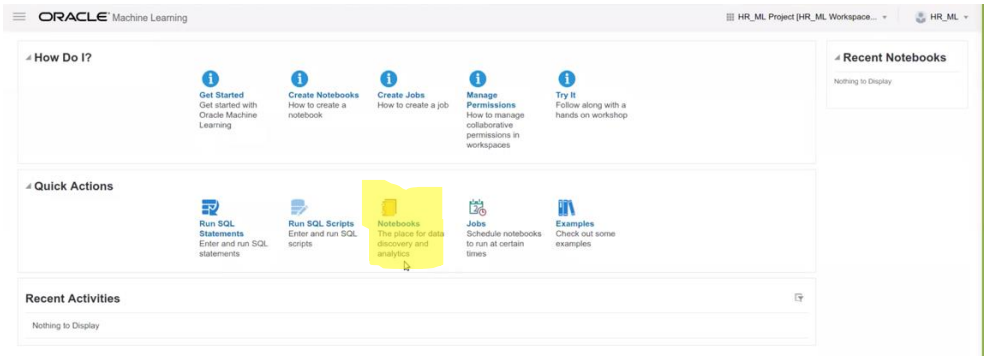
Vamos a la parte de desarrollo (Development) dentro de la consola de servicio, y pulsamos en "**Oracle Machine Learning Notebooks**":



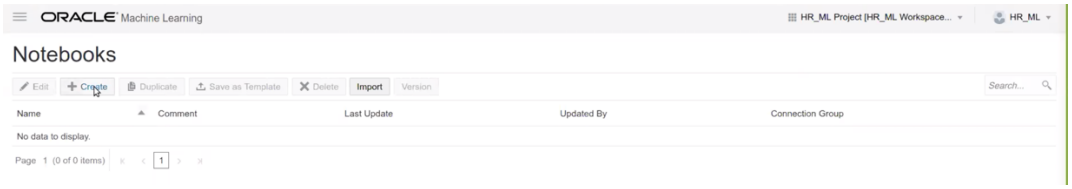
En la pantalla de login, nos conectamos con el usuario ML_HR/Autonomous#2020:



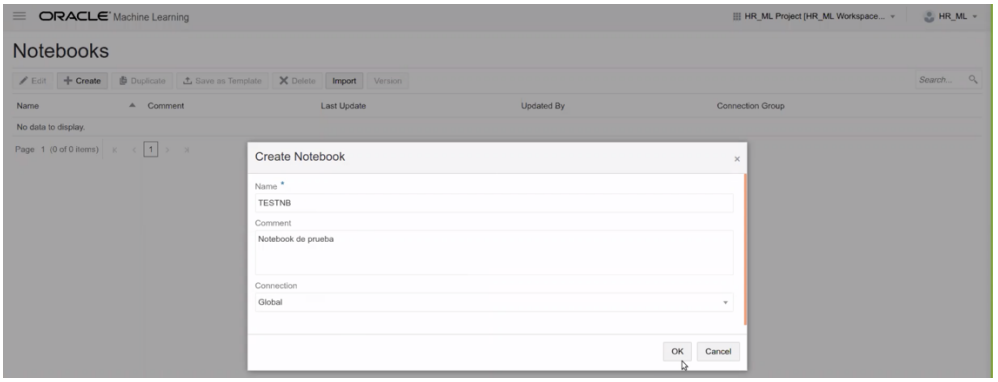
A continuación, aparece la pantalla principal de la sección de Machine Learning, elegimos la opción "**Notebooks**":



Esto dará paso a la creación de nuestro primer Notebook de Machine Learning. Pulsamos en el botón de crear:

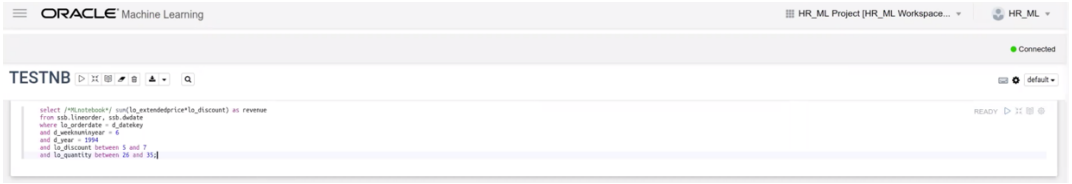


Damos un nombre al nuevo Notebook, en este caso TESTNB:

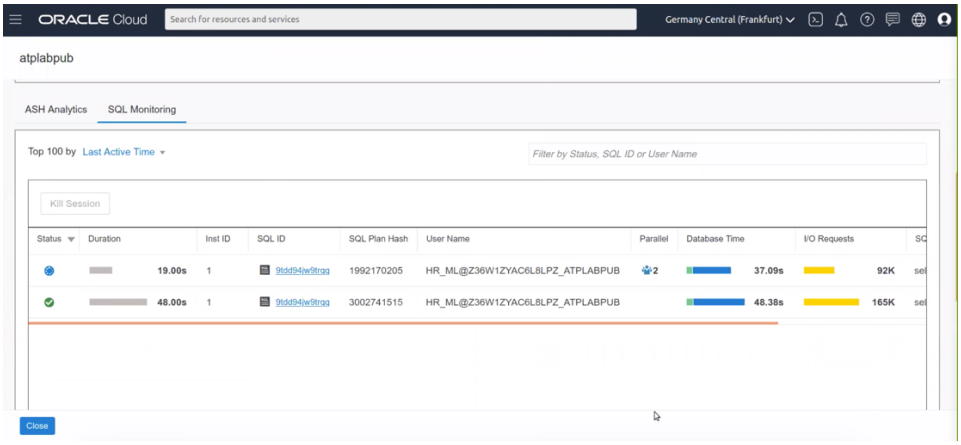


A continuación, ejecutamos una query en el nuevo notebook:

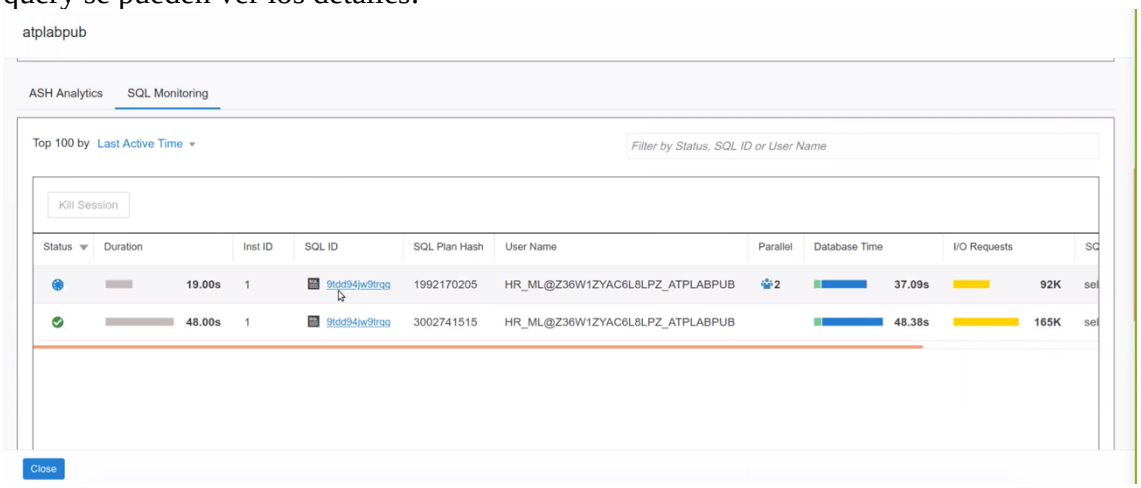
```
select /*MLnotebook*/ sum(lo_extendedprice*lo_discount) as revenue
from ssb.lineorder, ssb.dwdate
where lo_orderdate = d_datekey
and d_weeknuminyear = 6
and d_year = 1994
and lo_discount between 5 and 7
and lo_quantity between 26 and 35;
```



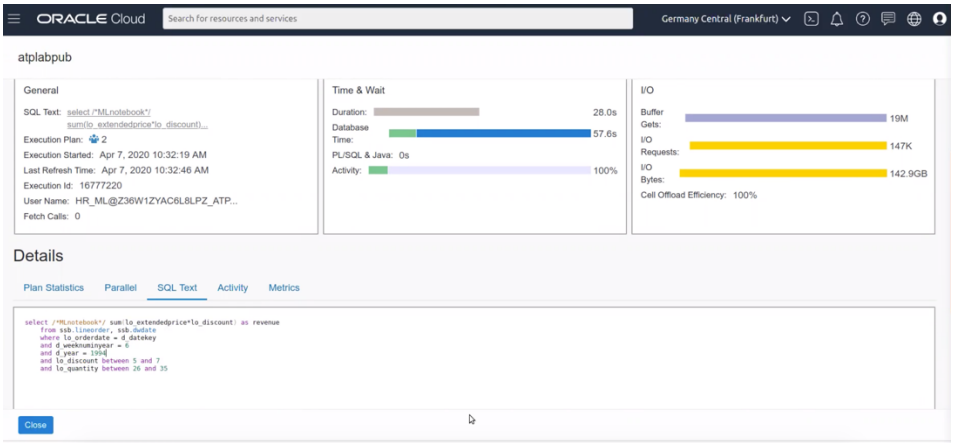
Podemos monitorizar la ejecución de la query en el “Performance Hub” desde la pestaña de ATP, dos pestañas a la izquierda de aquí:



Dentro de la pestaña SQL Monitoring, podemos ver la query ejecutada. Si entramos dentro de esta query se pueden ver los detalles:



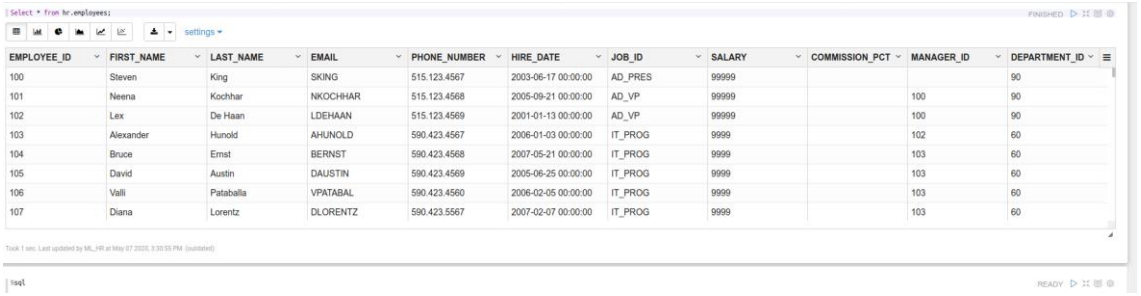
Podemos ver en la pestaña SQL Text que, en este caso, el motor de Machine Learning no ha reescrito la query. Tambien podemos ver los detalles asociados a esta query, como el plan de ejecución, estadísticas, actividad, métricas, etc



Finalmente volvemos al Notebook y comprobamos el resultado de la query:

Vamos a ejecutar ahora una query contra la tabla HR.employees desde el mismo Notebook:

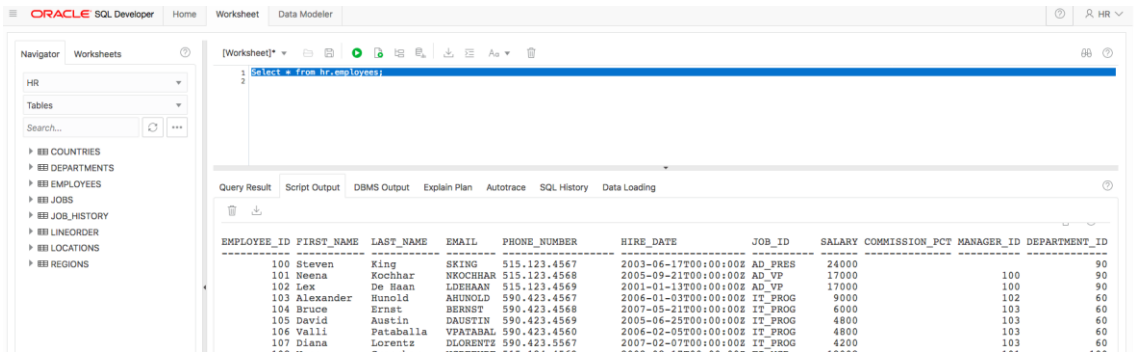
```
Select * from hr.employees;
```



EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
100	Steven	King	SKING	515.123.4567	2003-06-17 00:00:00	AD_PRES	99999			90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	2005-09-21 00:00:00	AD_VP	99999		100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	2001-01-13 00:00:00	AD_VP	99999		100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	2006-01-03 00:00:00	IT_PROG	9999		102	60
104	Bruce	Ernst	BERNST	590.423.4568	2007-05-21 00:00:00	IT_PROG	9999		103	60
105	David	Austin	DAUSTIN	590.423.4569	2005-06-25 00:00:00	IT_PROG	9999		103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	2006-02-05 00:00:00	IT_PROG	9999		103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	2007-02-07 00:00:00	IT_PROG	9999		103	60

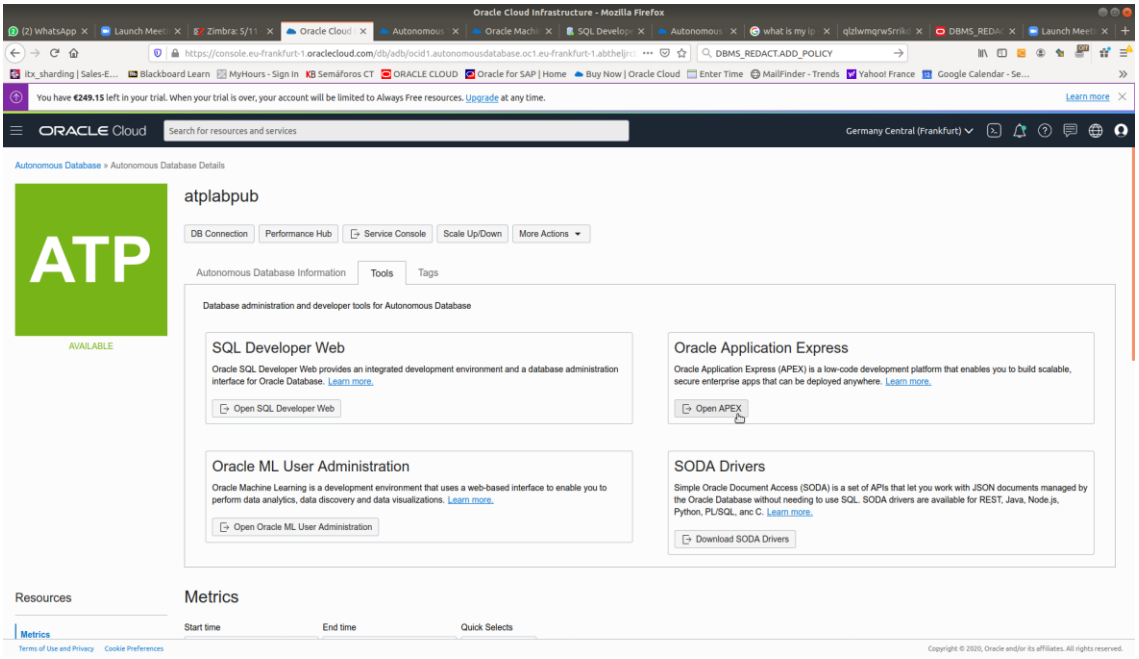
Observamos que el campo “salary” esta enmascarado con “9”, ocultando el valor real del campo en todas las filas. Esto es el efecto de la política de Data Redaction que hemos implementado anteriormente.

Podemos compararlo con la consulta que se hace con el usuario HR desde SQL Developer web.

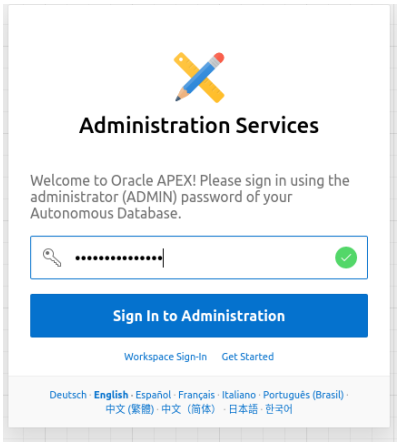


Ejercicio 4: Utilización de APEX

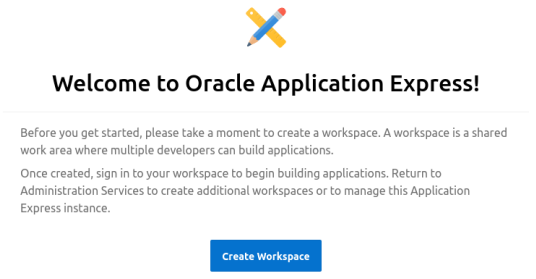
En el ejercicio siguiente, vamos a utlizar APEX. Desde la pantalla principal del ATP, en la pestaña "Tools", elegimos "Oracle Application Express":

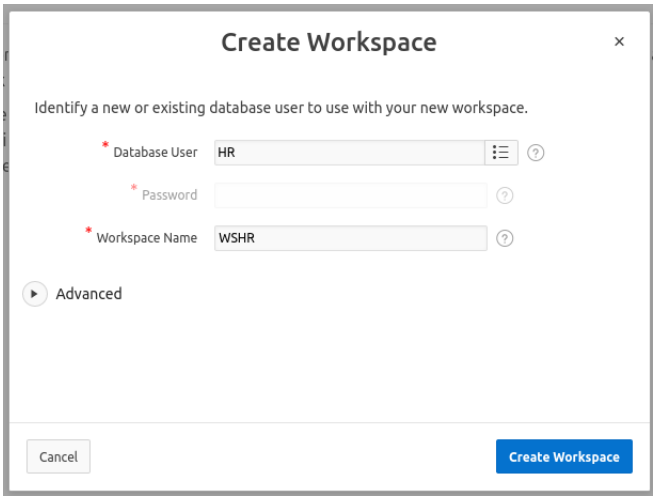


Primero nos conectamos con el usuario ADMIN a la consola de administración de APEX:

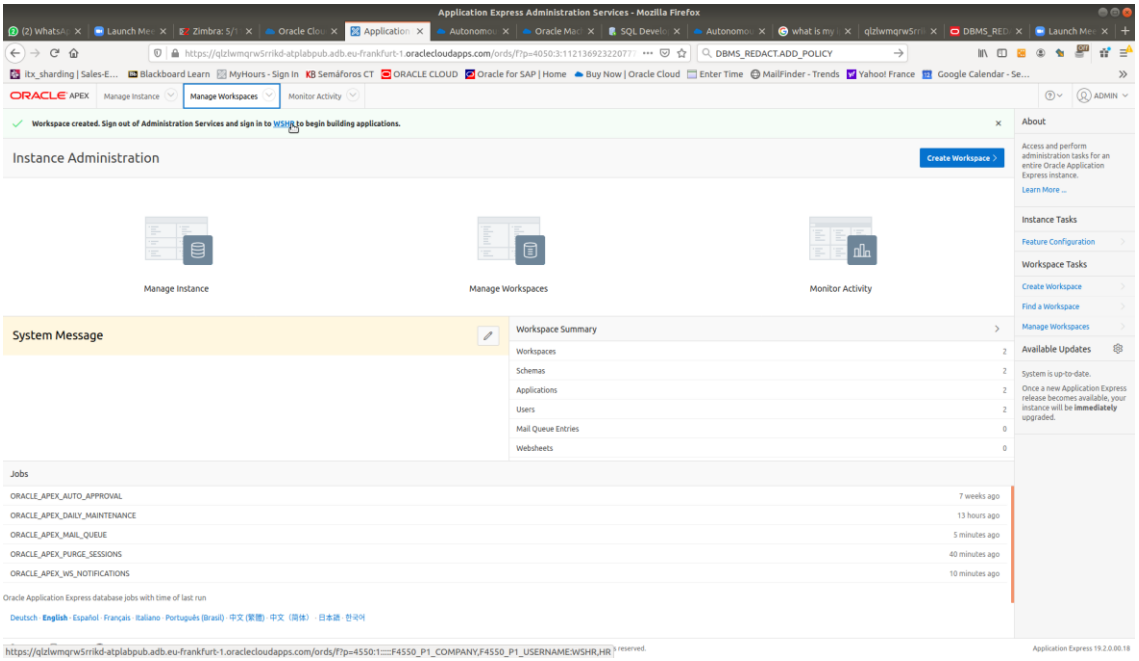


A continuación, creamos un "Workspace" con nombre WSHR para el usuario "HR"

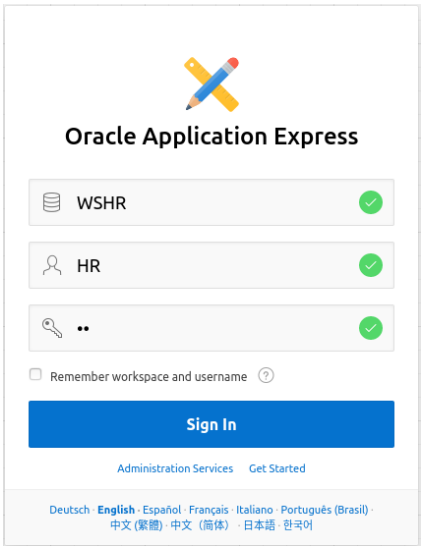




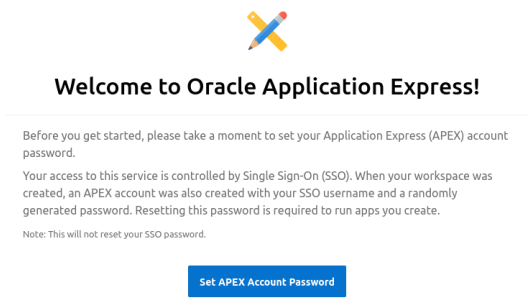
Una vez creado el Workspace, nos conectamos a APEX con el usuario HR, siguiendo el enlace arriba a la izquierda en la pantalla principal de APEX.



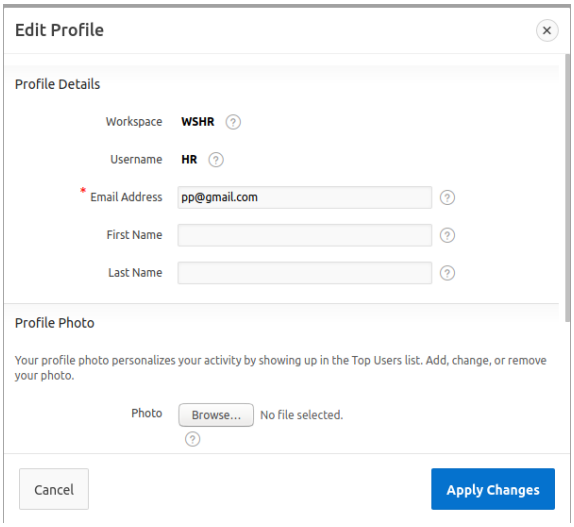
Y nos conectamos como HR/hr o hr/hr (**contraseña siempre en minúsculas**):



Seguimos los pasos siguientes:

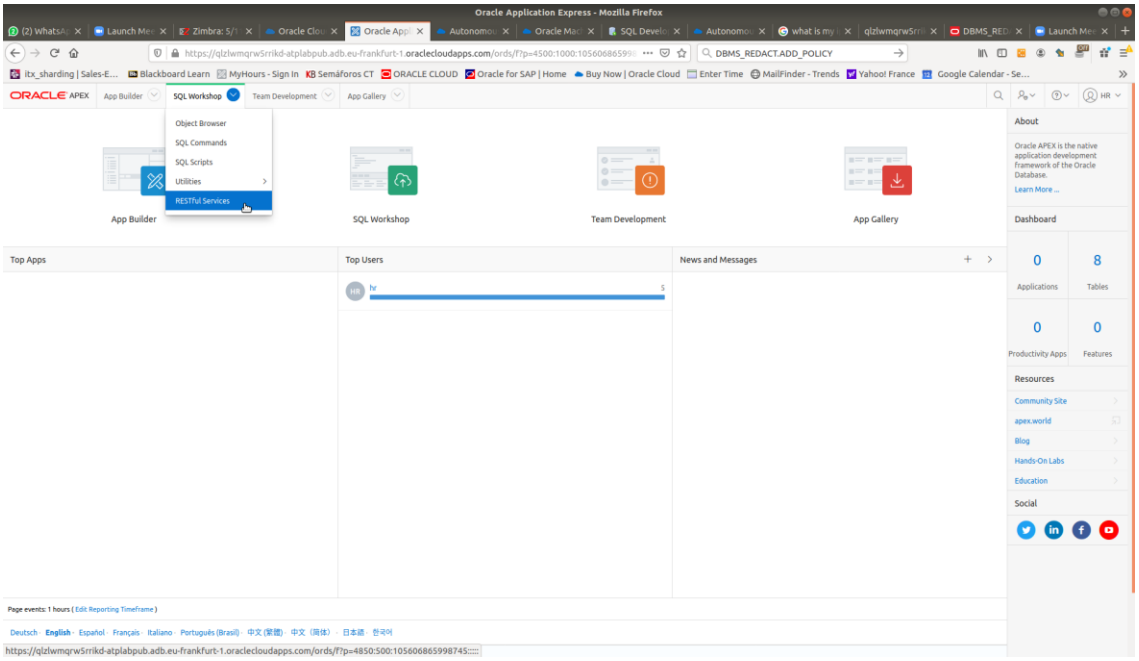


Completamos el perfil del usuario HR:

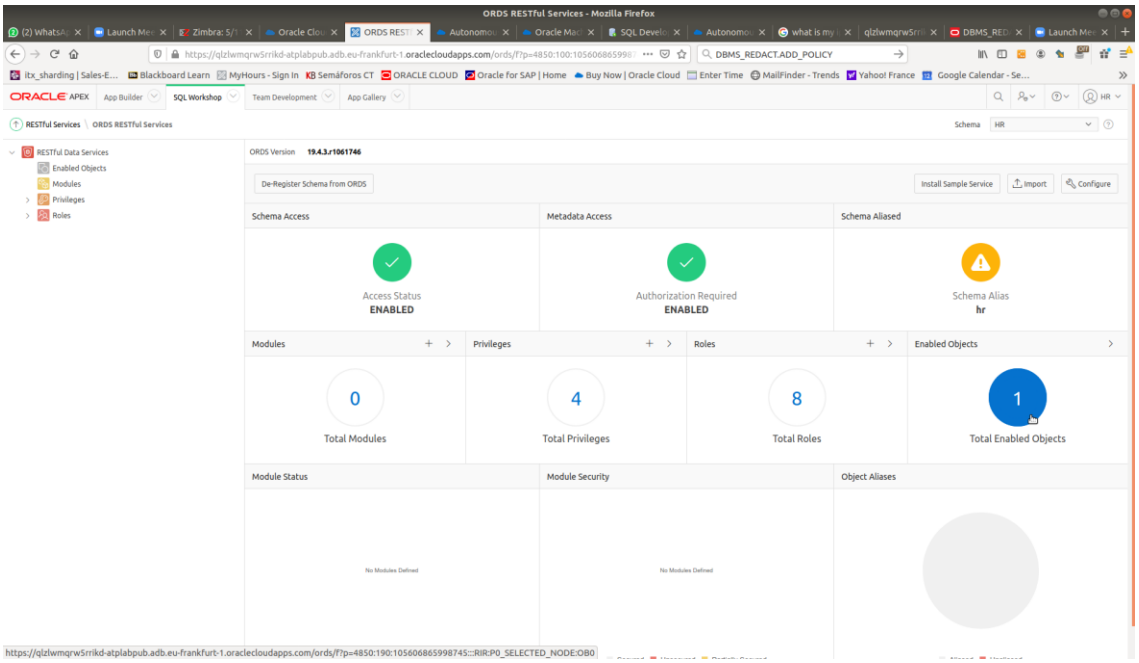


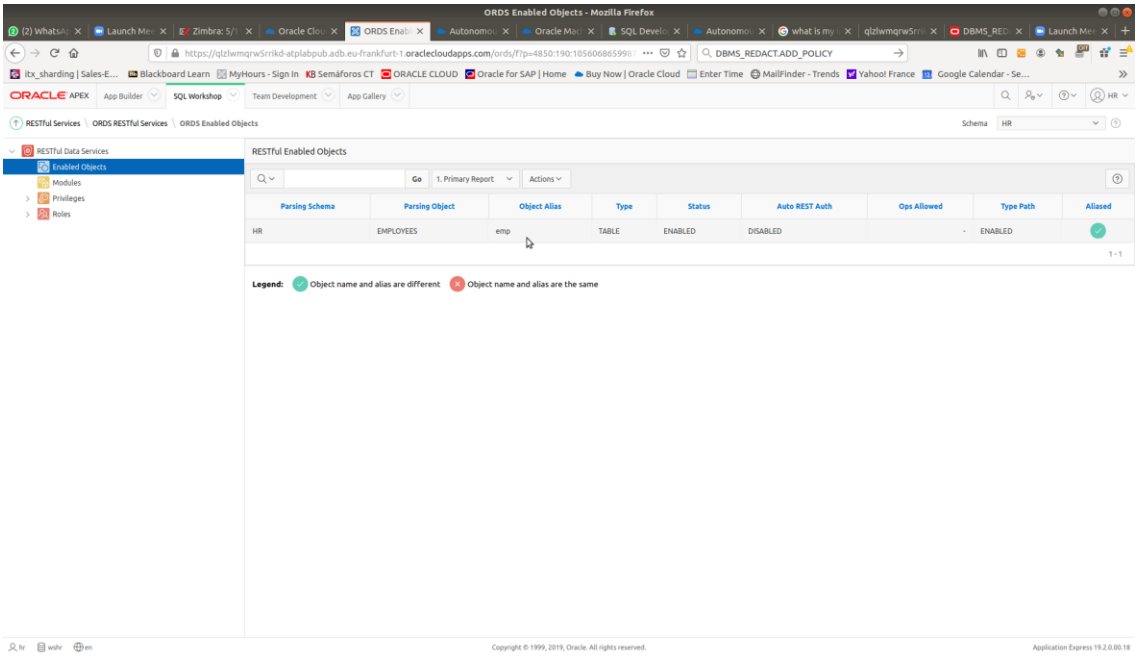
Esto nos lleva a la pantalla principal del workspace, desde donde podremos crear aplicaciones nuevas, gestionar el acceso por REST, etc ...

En esta pantalla, Pulsamos sobre el menu “SQL Workshop”, opción “Restful Services”:

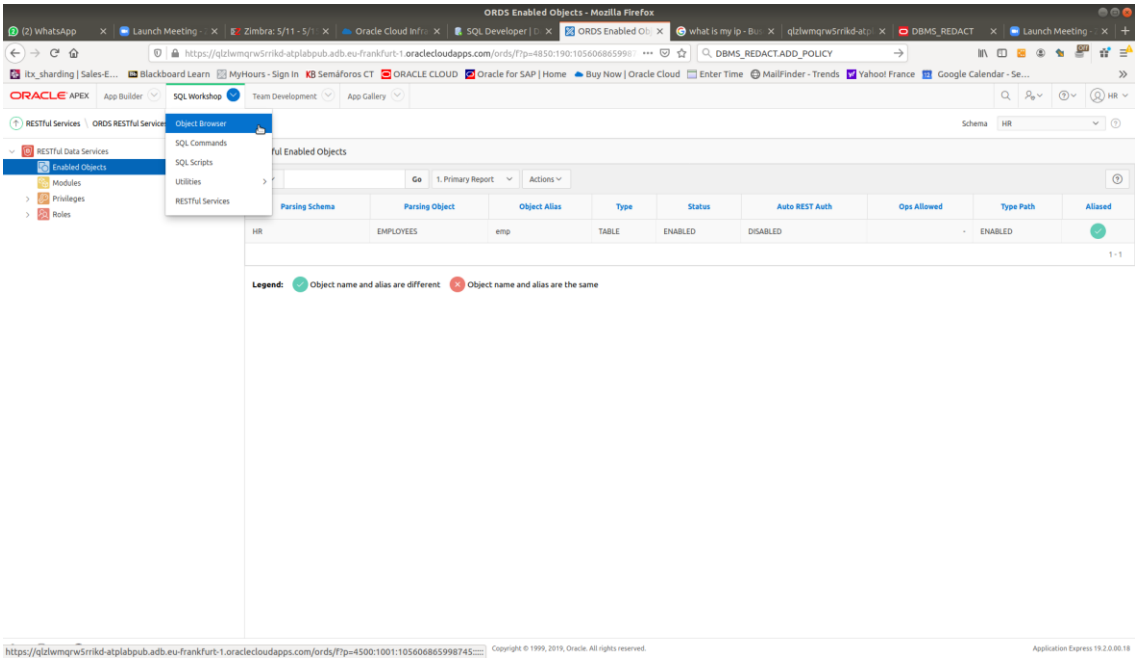


Vemos que ORDS está habilitada sobre el esquema HR, que su alias es “hr”, y que tiene un objeto habilitado para REST. Pulsamos sobre “Total Enabled Objects”:

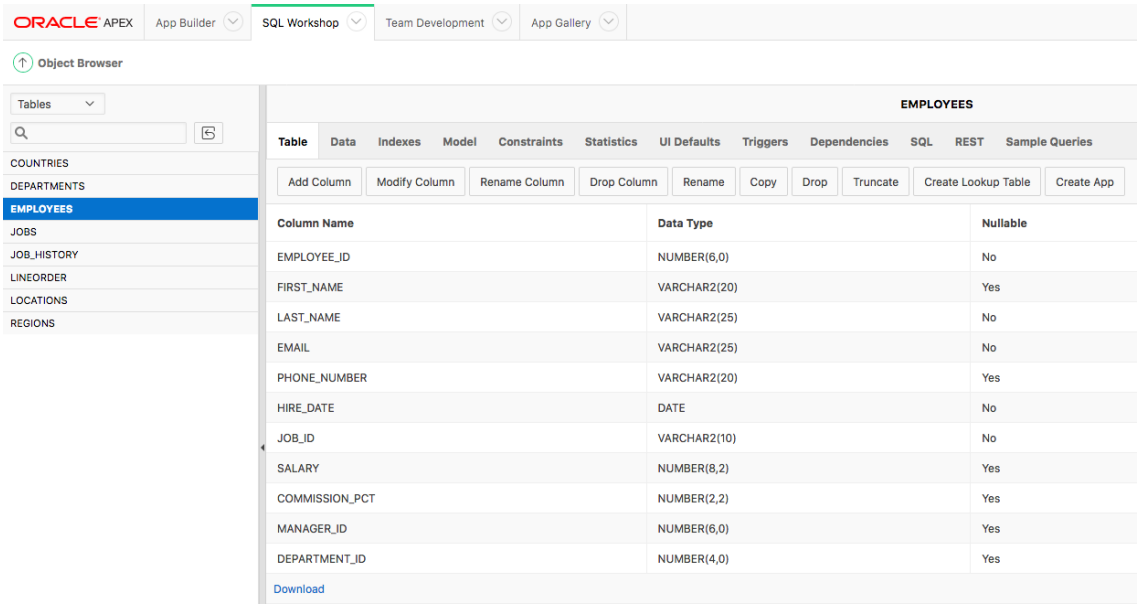




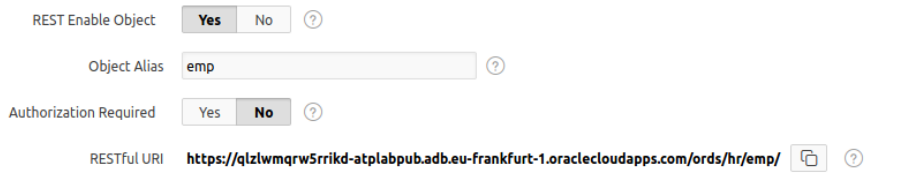
Ahora volvemos al menú SQL Workshop, y elegimos la opción “Object Browser”:



Esto nos lleva a una pantalla donde vemos los objetos del esquema HR. Pulsamos en el objeto "EMPLOYEES", y accedemos a la pestaña REST.



Aquí vemos la URL a utilizar para acceder a la tabla mediante API REST:



Si copiamos esta URL y la pegamos en un navegador, vemos los datos de la tabla, al igual que en un ejercicio anterior. Alternativamente, desde cualquiera de las máquinas "bastion", podemos acceder a esta URL mediante cURL:

```
curl https://qlzlwmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/100
```

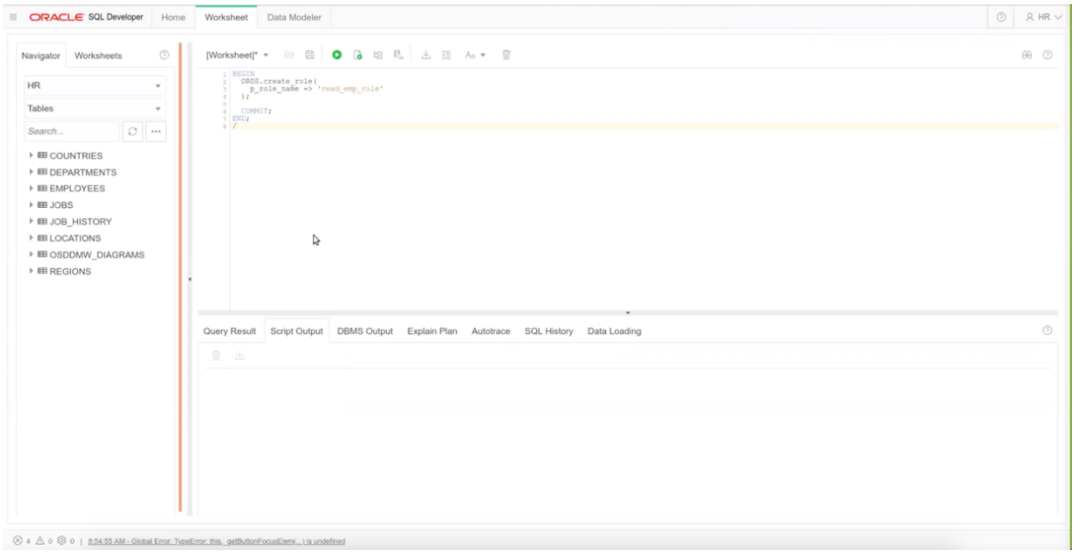
```
{"employee_id":100,"first_name":"Steven","last_name":"King","email":"SKING","phone_number":"515.123.4567","hir
e_date":"2003-06-
17T00:00:00Z","job_id":"AD_PRES","salary":24000,"commission_pct":null,"manager_id":null,"department_id":90,"li
nks":[{"rel":"self","href":"https://qlzlwmqrw5rrikd-atplabpub.adb.eu-frankfurt-
1.oraclecloudapps.com/ords/hr/emp/100"},{"rel":"edit","href":"https://qlzlwmqrw5rrikd-atplabpub.adb.eu-
frankfurt-1.oraclecloudapps.com/ords/hr/emp/100"},{"rel":"describedby","href":"https://qlzlwmqrw5rrikd-
atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/metadata-
catalog/emp/item"},{"rel":"collection","href":"https://qlzlwmqrw5rrikd-atplabpub.adb.eu-frankfurt-
1.oraclecloudapps.com/ords/hr/emp/"}]}
```

Ejercicio 5 (opcional): Configuración de seguridad de acceso OAuth2

En este ejercicio se explica como configurar seguridad de autenticación para acceder a los datos a través de REST API con un token de autenticación. Vamos a dotar el acceso a la tabla “employes” de seguridad mediante autenticación por token.
Nos conectamos al Sql*Developer Web como usuario HR, igual que en el ejercicio 2.

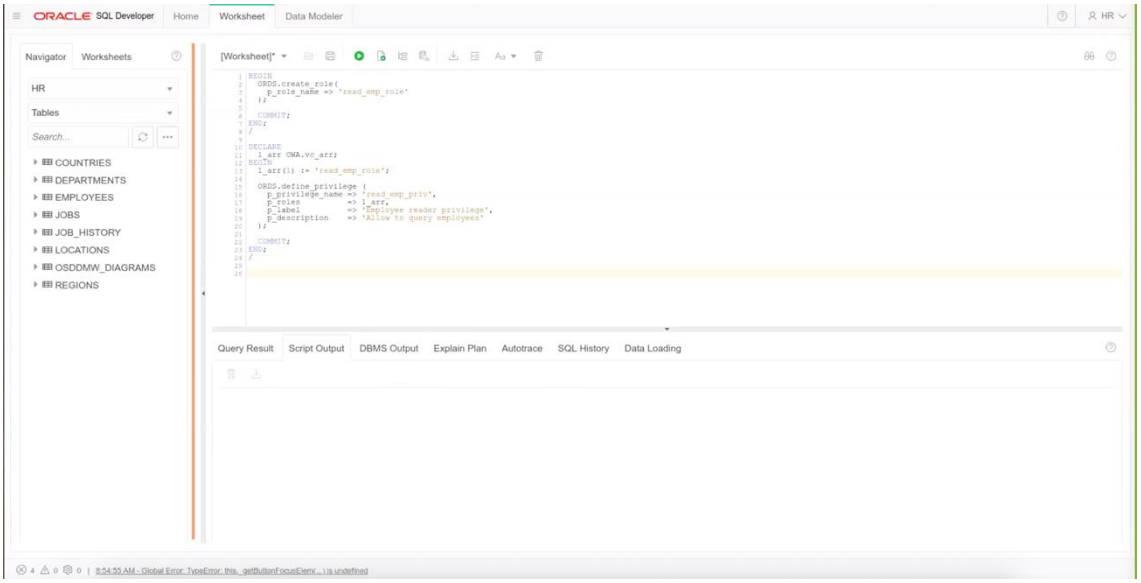
En primer lugar, hay que crear un rol, que se asociará al usuario HR y nos permitirá acceder al endpoint “/emp”:

```
BEGIN
  ORDS.create_role(
    p_role_name => 'read_emp_role'
  );
  COMMIT;
END;
/
```



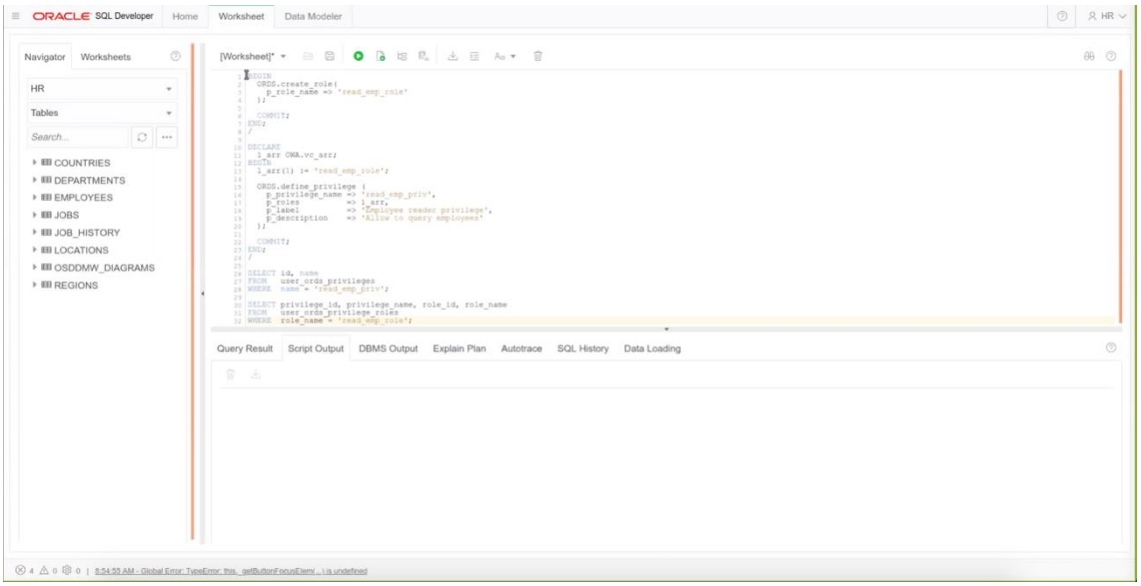
A continuación, se crea un privilegio en ORDS. Este privilegio lo asociamos al role creado en el paso anterior:

```
DECLARE
  l_arr OWA.vc_arr;
BEGIN
  l_arr(1) := 'read_emp_role';

  ORDS.define_privilege (
    p_privilege_name => 'read_emp_priv',
    p_roles          => l_arr,
    p_label          => 'Employee reader privilege',
    p_description    => 'Allow to query employees'
  );
  COMMIT;
END;
/
```



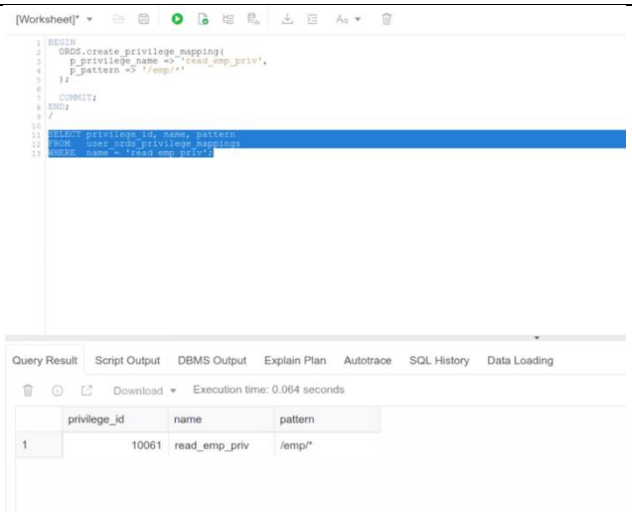
Con las siguientes queries, comprobamos que el rol ha sido correctamente asociado al privilegio de ORDS:

```
SELECT id, name
FROM   user_ords_privileges
WHERE  name = 'read_emp_priv';

SELECT privilege_id, privilege_name, role_id, role_name
FROM   user_ords_privilege_roles
WHERE  role_name = 'read_emp_role';
```

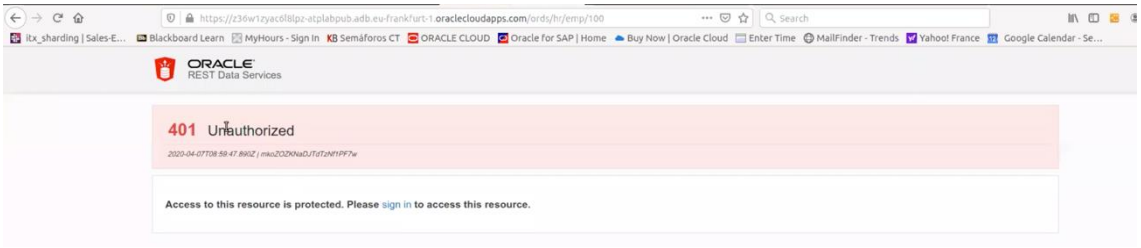


A continuación, mapeamos el privilegio a todas las terminaciones del endpoint “/emp/*”.

```
BEGIN
  ORDS.create_privilege_mapping(
    p_privilege_name => 'read_emp_priv',
    p_pattern => '/emp/*'
  );
  COMMIT;
END;
/

SELECT privilege_id, name, pattern
FROM   user_ords_privilege_mappings
WHERE  name = 'read_emp_priv';
```



Comprobamos si podemos acceder a los datos de la tabla employees a través de un navegador, con la URL utilizada en ejercicios anteriores:



La URL falla, por fallo de autorización.

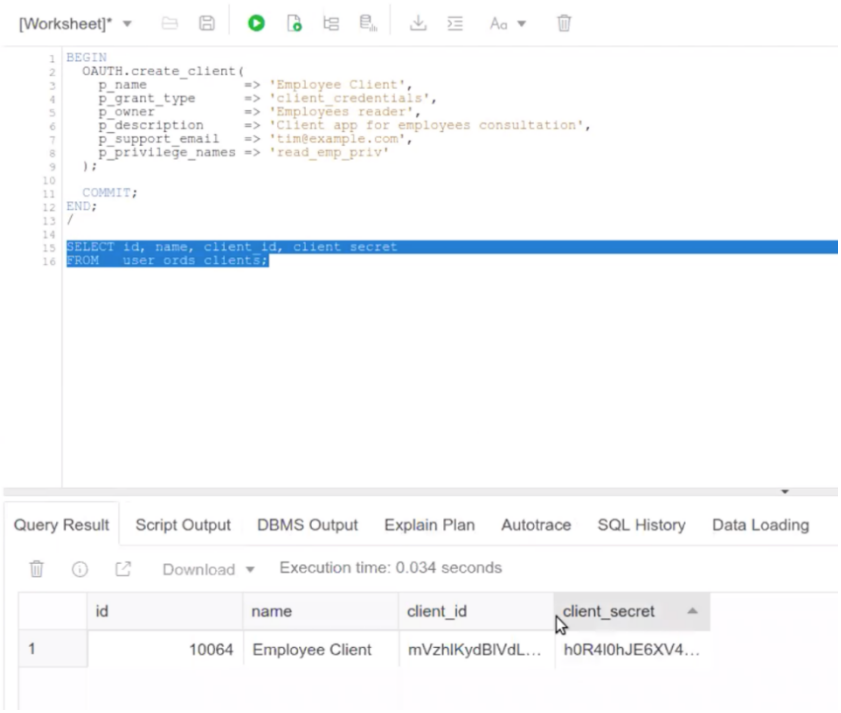
El siguiente paso es crear un token de autorización, valido durante una hora desde su ultima utilización.

Ejecutamos lo siguiente:

```
BEGIN
  OAUTH.create_client(
    p_name            => 'Employee Client',
    p_grant_type      => 'client_credentials',
    p_owner           => 'Employees reader',
    p_description     => 'Client app for employees consultation',
    p_support_email   => 'tim@example.com',
    p_privilege_names => 'read_emp_priv'
  );
  COMMIT;
END;
/
```

```
SELECT id, name, client_id, client_secret
FROM   user_ords_clients;
```

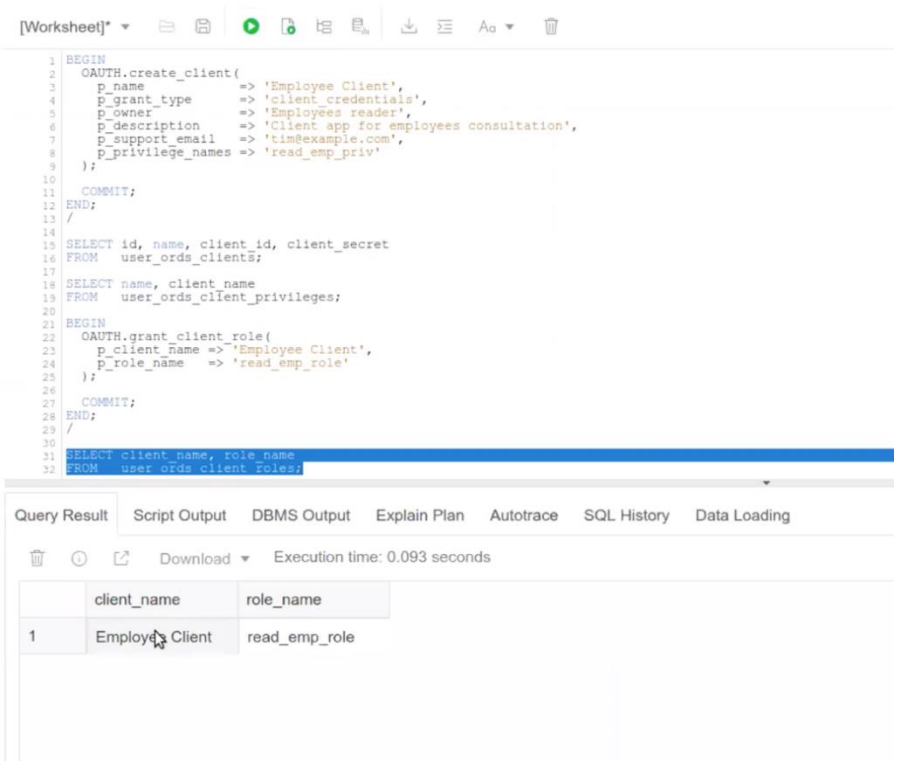
La última sentencia SELECT devuelve un client ID y client secret. Los copiamos en un fichero de texto para introducirlos en la próxima llamada REST para conseguir un token de autenticación.



A continuación, mapeamos el token de autenticación con el rol que se ha creado anteriormente para la tabla de empleados:

```
BEGIN
  OAUTH.grant_client_role(
    p_client_name => 'Employee Client',
    p_role_name   => 'read_emp_role'
  );
  COMMIT;
END;
/
```

```
SELECT client_name, role_name
FROM   user_ords_client_roles;
```



El siguiente paso es comprobar mediante el comando “curl”, desde una terminal Linux, si se puede acceder introduciendo el token de autenticación. Esto lo podemos hacer desde cualquiera de las máquinas de bastion.

Si no hemos apuntado el client_id y el secret en uno de los pasos anteriores, lo podemos consultar de nuevo con la siguiente query:

```
SELECT id, name, client_id, client_secret
FROM   user_ords_clients;
```

A continuación, ejecutamos el siguiente comando, desde cualquiera de las máquinas bastion, para conseguir un token. Observamos en el comando de cURL el uso del parametro “**--user**”, con el valor **<client_id>:<secret>**:
La URL de oAUTH es nuestro REST Endpoint ya utilizado en varios ejercicios anteriores, completado por “**/hr/oauth/token**”:

```
#CLIENT_ID     : x3n1g7heGXI0zxN_DJrIXw..
#CLIENT_SECRET : Az4WOTviFaDjgHgSMq-KLg..
#OAUTH URL     : https://z36w1zyac6l8lpz-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/oauth/token

curl -i -k --user x3n1g7heGXI0zxN_DJrIXw..:Az4WOTviFaDjgHgSMq-KLg.. --data "grant_type=client_credentials"
https://<use_su_ATP_ORDS_URL>/ords/hr/oauth/token
```

El comando anterior nos devuelve un token, que utilizamos ahora para consultar la tabla “employees”: (sustituir <TOKEN> por el token devuelto por el paso anterior):

```
curl -i -k -H"Authorization: Bearer <TOKEN>" https://z36w1zyac6l8lpz-atplabpub.adb.eu-frankfurt-
1.oraclecloudapps.com/ords/hr/emp/100
```