

HOL5 - ATP tooling

INDICE

HOLS - ATP TOOLING	1
EJERCICIO 1: CREAR USUARIO DE MACHINE LEARNING.....	3
EJERCICIO 2: UTILIZAR SQL*DEVELOPER WEB.....	5
EJERCICIO 3: CREAR UN NOTEBOOK EN ORACLE MACHINE LEARNING.....	16
EJERCICIO 4: UTILIZACIÓN DE APEX.....	20
EJERCICIO 5 (OPCIONAL): CONFIGURACIÓN DE SEGURIDAD DE ACCESO OAUTH2.....	26



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**”

The screenshot shows the 'Autonomous Database Details' page for the database 'atplabpub'. At the top, there are tabs for 'DB Connection', 'Performance Hub', 'Service Console', 'Scale Up/Down', and 'More Actions'. Below these, under 'Autonomous Database Information', the 'Tools' tab is selected. A red box highlights the 'Open Oracle ML User Administration' button, which is located under the 'Oracle ML User Administration' section. This section also includes a brief description of Oracle Machine Learning and links to 'Open SQL Developer Web' and 'Open APEX'.

En la pantalla de login, entrar las **credenciales** de **ADMIN**:

Usuario: ADMIN

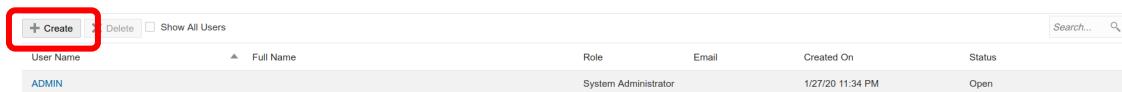
Contraseña: Autonomous#2020

The screenshot shows the 'SIGN IN' page for the Oracle Machine Learning Database Administrator. It features a large blue header with a cloud icon and the word 'SIGN IN'. Below the header, it says 'Database name: ATPLABPUB'. The main form asks 'Sign in with your Oracle Machine Learning Database Administrator credentials'. It has fields for 'USERNAME *' (containing 'ADMIN') and 'PASSWORD *' (containing a masked password). A red box highlights the 'Sign In' button at the bottom of the form.

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



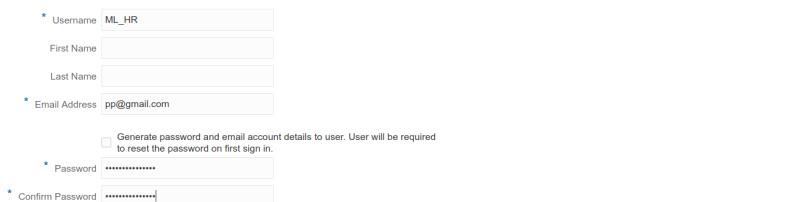
Users



User Name	Full Name	Role	Email	Created On	Status
ADMIN		System Administrator		1/27/20 11:34 PM	Open

Cliquear el botón “Create”.

Create User



Username: ML_HR
First Name:
Last Name:
Email Address: pp@gmail.com
 Generate password and email account details to user. User will be required to reset the password on first sign in.
Password:
Confirm Password:

En la pantalla de creación de usuario, llenar la información y pulsar el botón “Create”.

Usuario: ML_HR

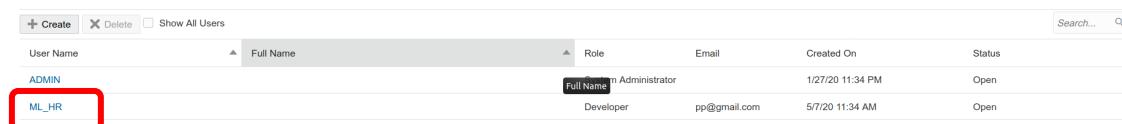
Password: Autonomous#2020

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:

Users



User Name	Full Name	Role	Email	Created On	Status
ADMIN		System Administrator		1/27/20 11:34 PM	Open
ML_HR		Developer	pp@gmail.com	5/7/20 11:34 AM	Open



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”.

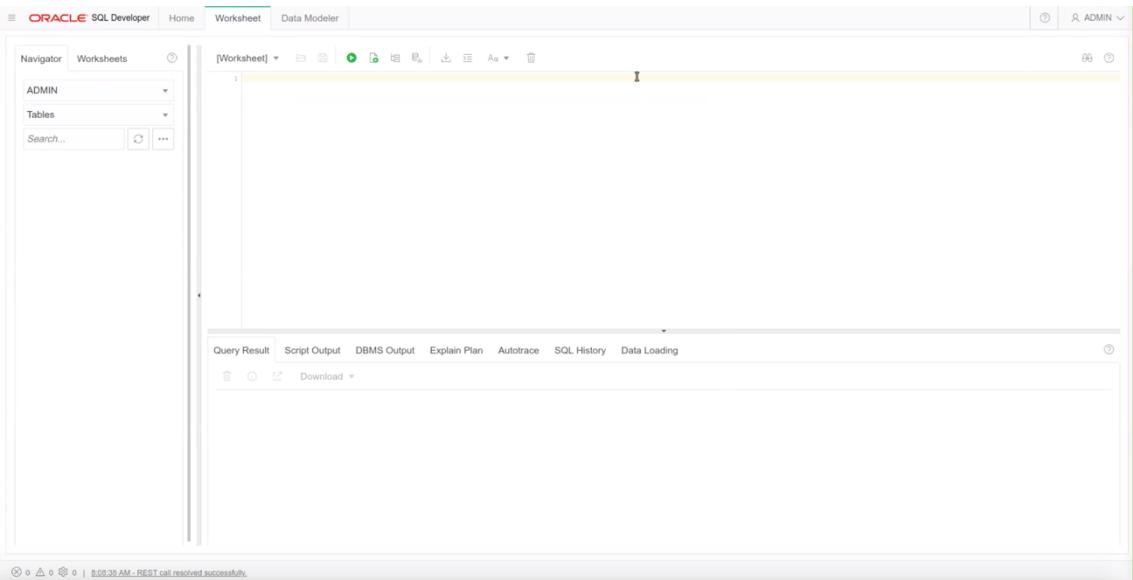
The screenshot shows the 'Autonomous Database Details' page for 'atplabpub'. At the top, there's a green box labeled 'ATP' with 'AVAILABLE' underneath. Below it, there are several tabs: 'DB Connection', 'Performance Hub', 'Service Console', 'Scale Up/Down', and 'More Actions'. The 'Tools' tab is currently selected and highlighted with a red box. Under the 'Tools' tab, there are four sections: 'SQL Developer Web' (with a red box around the 'Open SQL Developer Web' button), 'Oracle Application Express' (with a 'Open APEX' button), 'Oracle ML User Administration' (with a 'Open Oracle ML User Administration' button), and 'SODA Drivers' (with a 'Download SODA Drivers' button).

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

The screenshot shows the login screen for 'SQL Developer Web'. At the top left is the Oracle logo and the text 'SQL Developer Web'. Below that is a form with two fields: 'Username' (containing 'ADMIN') and 'Password' (an empty field). At the bottom of the form is a large blue 'Sign in' button.

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
    ord_s_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;  
/
```

[Worksheet]*

```
1 BEGIN  
2 SYS.DBMS_REDACT.ADD_POLICY(  
3 object_schema=> 'HR$',  
4 object_name => 'EMPLOYEES',  
5 column_name => 'SALARY',  
6 column_description => 'Employee salary',  
7 policy_name => 'POL_HIDE_SALARY',  
8 policy_description => 'Hide salary',  
9 function_type => DBMS_REDACT.PARTIAL,  
10 function_parameters => '9,1,9',  
11 expression => 'SYS_CONTEXT(''USERENV'', ''SESSION_USER'') = ''ML_HR''');  
12 end;  
13 /
```

Query Result

Script Output

DBMS Output

Explain Plan

Autotrace

SQL Histor



PL/SQL procedure successfully completed.

Elapsed: 00:00:00.311

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.



The screenshot shows the Oracle SQL Developer interface. In the top right corner, there is a dropdown menu with the user 'ADMIN' and a 'Sign Out' option highlighted. Below the menu, there are links for 'Preferences', 'Log', 'About', and 'Sign Out'. The main workspace contains a code editor with the following PL/SQL script:

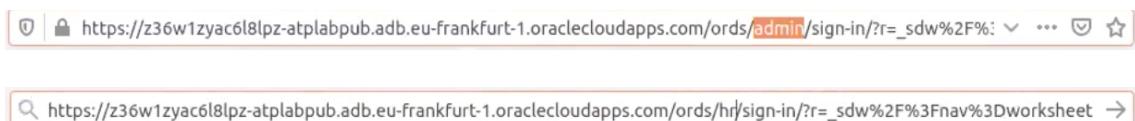
```

1  BEGIN
2      ords_admin.enable_schema (
3          p_enabled          => TRUE,
4          p_url_mapping_type => 'BASE_PATH',
5          p_auto_rest_auth   => 'TRUE',
6          p_auto_rest_d-auth => TRUE
7      );
8  END;
9  /

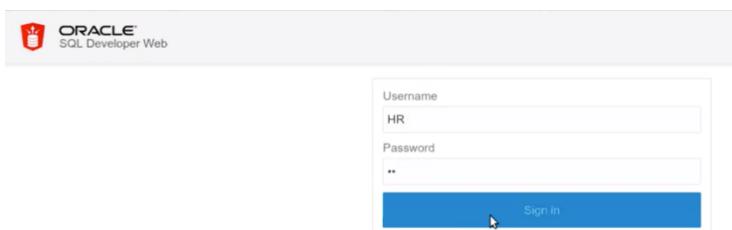
```

Below the code editor, the status bar displays 'PL/SQL procedure successfully completed.' and 'Elapsed: 00:00:00.409'.

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”**:

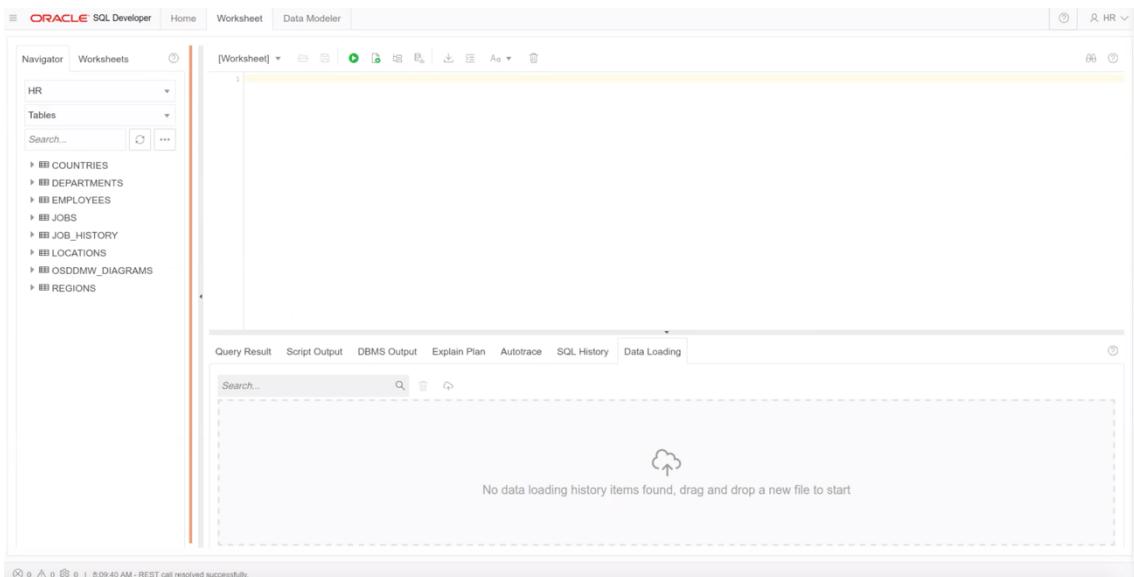


Volverá a aparecer la consola de login, volvemos a introducir el nombre de usuario y contraseña. En este caso el usuario HR/hr o hr/hr (**contraseña siempre en minúsculas**).



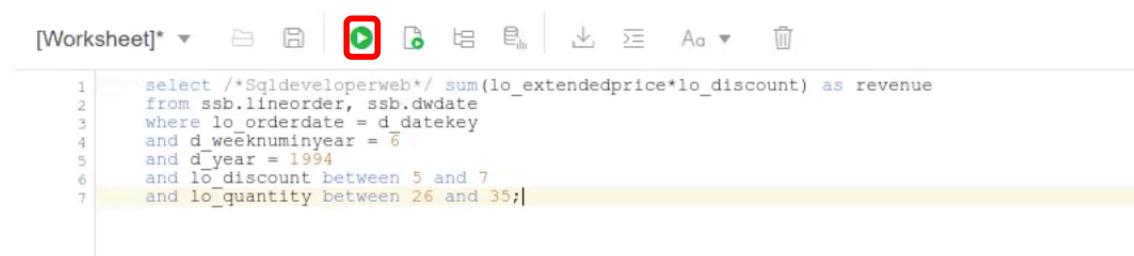
En la siguiente pantalla podemos ver la misma consola de SQL Developer Web, pero en este caso a la izquierda podemos ver las tablas del esquema HR:





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:



The screenshot shows the Oracle Database SQL Monitoring interface. It displays a session named '228g0dh6lrmz' with a duration of 20.00s. The session is executing a query with SQL ID 817007416, user HR@Z36W1ZYAC0L8LPZ_ATPLABPUB, and a database time of 20.40s. The query is a 'select' statement. The interface includes tabs for Kill Session, Status, Duration, Init ID, SQL ID, SQL Plan Hash, User Name, Parallel, Database Time, I/O Requests, and SQL Text.

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

The screenshot shows the Oracle Database Real-time SQL Monitoring interface for session '228g0dh6lrmz'. The query text is: 'select * from (select q.* , row_number() over (order by 1) RN from ...) where RN between 11 and 12'. The duration is 28.0s, and the database time is 28.6s. The activity is 100%. The interface includes tabs for Back, Overview, General, Time & Wait, and I/O. It also has Save Report and Refresh buttons.

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

The screenshot shows the Oracle Database SQL Developer Web interface. The SQL Text tab is selected, displaying the query code: 'select q.* , row_number() over (order by 1) RN from ... where RN between 11 and 12'. Other tabs include Plan Statistics, Activity, and Metrics.

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



The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, the 'Worksheet' tab is selected. On the left, the Navigator pane shows the HR schema with tables like COUNTRIES, DEPARTMENTS, EMPLOYEES, JOBS, JOB_HISTORY, LOCATIONS, OSDDMW_DIAGRAMS, and REGIONS. The main workspace contains a query in the [Worksheet] tab:

```
1 select * from employees e
  2   join departments d on e.department_id = d.department_id
  3   join locations l on d.location_id = l.location_id
  4   join countries c on l.country_id = c.country_id
  5   join job_history jh on e.employee_id = jh.employee_id
  6   join jobs j on j.job_id = jh.job_id
  7 where d.work_location = 'TURKEY'
    and l.country_id between 3 and 7
    and jh.end_date between '2000-01-01' and '2000-06-30'
```

The results are displayed in the 'Query Result' tab, showing a single row with the value 'revenue' and a value of 26113.

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;
```

The screenshot shows the Oracle SQL Developer interface with a yellow highlight on the query code. The [Worksheet] tab is selected, and the code is identical to the one above:

```
1 select json_object
  2 (
  3   KEY 'ID' is E.employee_id,
  4   KEY 'full_name' is E.first_name || ' ' || E.last_name,
  5   KEY 'email' is E.email,
  6   KEY 'phone' is E.phone_number
  7 ) as EMPLOYEE_PAYLOAD
  8 from employees E;
```

Podemos ver el resultado en formato JSON:



```

1 select json_object
2   ( KEY 'id' IS E.employee_id,
3     KEY 'full_name' IS E.first_name || ' ' || E.last_name,
4     KEY 'email' IS E.email,
5     KEY 'phone' IS E.phone_number
6   ) AS EMPLOYEE
7   FROM employees;

```

Query Result

```

[{"ID":201,"full_name":"Michael Hartstein","email":"MHARTSTEIN","phone":"515.123.5555"}, {"ID":202,"full_name":"Pat Fay","email":"PFAY","phone":"603.123.6666"}, {"ID":203,"full_name":"Susan Mavris","email":"SMARVIS","phone":"515.123.7777"}, {"ID":204,"full_name":"Hermann Baer","email":"HBAER","phone":"515.123.8888"}, {"ID":205,"full_name":"Shelley Higgins","email":"SHIGGINS","phone":"515.123.8080"}, {"ID":206,"full_name":"William Gietz","email":"WGIETZ","phone":"515.123.8181"}]

```

Elapsed: 00:00:00.029
107 rows selected.

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

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

```
1 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;
/

```

```

1 DECLARE
2   PRAGMA AUTONOMOUS_TRANSACTION;
3 BEGIN
4   ORDS.ENABLE_OBJECT(p_enabled => TRUE,
5                      p_schema => 'HR',
6                      p_object => 'EMPLOYEES',
7                      p_object_type => 'TABLE',
8                      p_object_alias => 'emp',
9                      p_auto_rest_auth => FALSE);
10  commit;
11 END;
12 /

```



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”:





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

itx_sharding | Sales-E... Blackboard Learn MyHours - Sign In KB Semáforos CT ORACLE CLOUD Oracle for SAP | Home

JSON Raw Data Headers

Save Copy Collapse All Expand All Filter JSON

```

▼ items:
  ▼ 0:
    employee_id: 100
    first_name: "Steven"
    last_name: "King"
    email: "SKING"
    phone_number: "515.123.4567"
    hire_date: "2003-06-17T00:00:00Z"
    job_id: "AD_PRES"
    salary: 24000
    commission_pct: null
    manager_id: null
    department_id: 90
  ▼ links:
    ▼ 0:
      rel: "self"
      href: "https://qlzlwmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/100"
  ▼ 1:
    employee_id: 101
    first_name: "Neena"
    last_name: "Kochhar"
    email: "NKOCHHAR"
    phone_number: "515.123.4568"
    hire_date: "2005-09-21T00:00:00Z"

```

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

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

itx_sharding | Sales-E... Blackboard Learn MyHours - Sign In KB Semáforos CT ORACLE CLOUD Oracle for SAP | Home

JSON Raw Data Headers

Save Copy Collapse All Expand All Filter JSON

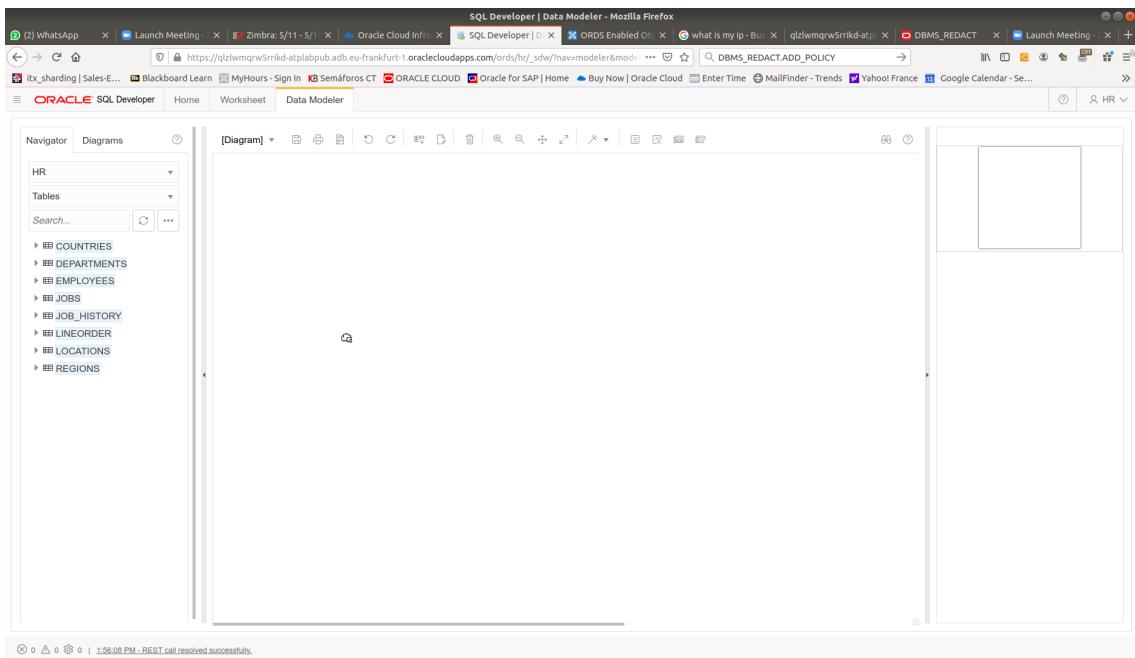
```

employee_id: 100
first_name: "Steven"
last_name: "King"
email: "SKING"
phone_number: "515.123.4567"
hire_date: "2003-06-17T00:00:00Z"
job_id: "AD_PRES"
salary: 24000
commission_pct: null
manager_id: null
department_id: 90
  ▼ links:
    ▼ 0:
      rel: "self"
      href: "https://qlzlwmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/100"
    ▼ 1:
      rel: "edit"
      href: "https://qlzlwmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/100"
    ▼ 2:
      rel: "describedby"
      href: "https://qlzlwmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/metadata-catalog/emp/item"
    ▼ 3:
      rel: "collection"
      href: "https://qlzlwmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/"

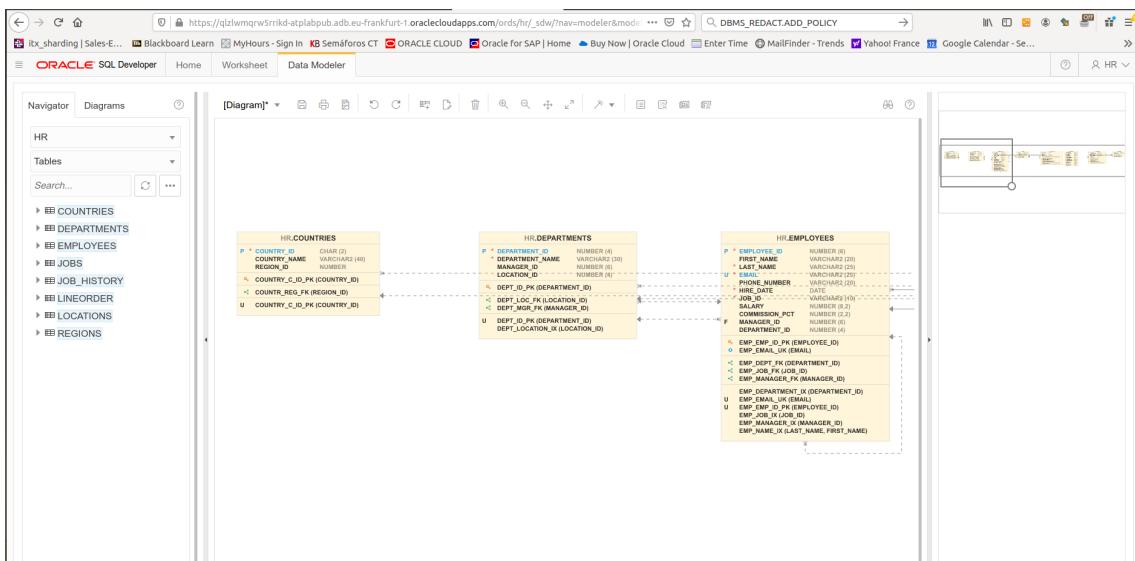
```

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**”:

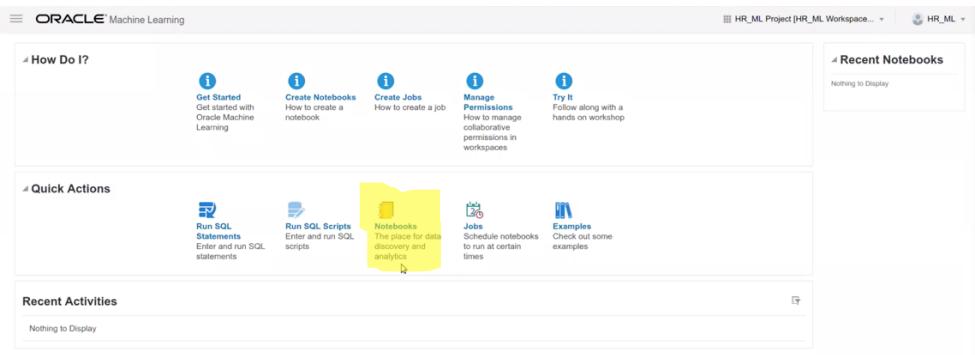
The screenshot shows the Oracle Cloud Infrastructure Service Console. On the left, there's a sidebar with 'Autonomous Transaction Processing' selected. Under 'DATABASE' (which is also selected), there are several options: 'Download Oracle Instant Client', 'Oracle APEX', 'Oracle Machine Learning Notebooks' (which is highlighted in red), and 'RESTful Services and SODA'. The 'Oracle Machine Learning Notebooks' section contains a brief description and a 'Copy URL' button.

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

The screenshot shows a sign-in page for Oracle Machine Learning. It asks for a 'Database name:' which is filled with 'ATPLABPUB'. Below it, it says 'Sign in with your Oracle Machine Learning Database User credentials'. There are fields for 'USERNAME *' containing 'ML_HR' and 'PASSWORD *' with a masked password. A blue 'Sign In' button is at the bottom.

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 pestanas a la izquierda de aquí:



Top 100 by Last Active Time

Status	Duration	Inst ID	SQL ID	SQL Plan Hash	User Name	Parallel	Database Time	I/O Requests	SC
Running	19.00s	1	9tdd94iw9tqg	1992170205	HR_ML@Z36W1ZYAC6L8LPZ_ATPLABPUB	2	37.09s	92K	sel
Running	48.00s	1	9tdd94iw9tqg	3002741515	HR_ML@Z36W1ZYAC6L8LPZ_ATPLABPUB		48.38s	165K	sel

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

Kill Session

Status	Duration	Inst ID	SQL ID	SQL Plan Hash	User Name	Parallel	Database Time	I/O Requests	SC
Running	19.00s	1	9tdd94iw9tqg	1992170205	HR_ML@Z36W1ZYAC6L8LPZ_ATPLABPUB	2	37.09s	92K	sel
Running	48.00s	1	9tdd94iw9tqg	3002741515	HR_ML@Z36W1ZYAC6L8LPZ_ATPLABPUB		48.38s	165K	sel

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

General

```
SQL Text: select /*ML.notebook*/
           sum(l_extendedprice*l_discount) as revenue
  from sub_lnlinesales, sub_order
 where lo_orderdate = d_datekey
 and s_suppregion = 'S'
 and d_year = 1994
 and l_discount between 5 and 7
 and l_quantity between 20 and 35
```

Time & Wait

Duration	28.0s
Database	57.6s
Time	
PL/SQL & Java	
Activity	100%

I/O

Buffer Gets	19M
I/O Requests	147K
I/O Bytes	142.9GB
Cell Offload Efficiency	100%

Details

- Plan Statistics
- Parallel
- SQL Text**
- Activity
- Metrics

```
select /*ML.notebook*/
           sum(l_extendedprice*l_discount) as revenue
  from sub_lnlinesales, sub_order
 where lo_orderdate = d_datekey
 and s_suppregion = 'S'
 and d_year = 1994
 and l_discount between 5 and 7
 and l_quantity between 20 and 35
```

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



The screenshot shows the Oracle Machine Learning Notebook interface. A query is being run against a database named 'TESTNB'. The query is:

```
select /*+materialize*/ sum(s.extendedprice*ls.discount) as revenue
from stb_lineorder stb_order
join stb_lineitem stb_order
and s.warehousekey = l.warehousekey
and s.orderkey = l.orderkey
and ls.discount between .5 and 1;

```

The result of the query is displayed in a table titled 'REVENUE' with one row:

REVENUE
2011319600347

At the bottom of the notebook, there is a status bar indicating 'Took 32 sec. Last updated by HR_ML at April 07 2020, 11:32:09 PM (continued)'.

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

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

The screenshot shows the results of the query 'Select * from hr.employees;' in the Oracle Machine Learning Notebook. The results are displayed in a table with the following columns: EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER, HIRE_DATE, JOB_ID, SALARY, COMMISSION_PCT, MANAGER_ID, and DEPARTMENT_ID. The data is as follows:

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	2003-06-17T00:00:00Z	AD_PRES	99999			90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	2005-09-21T00:00:00Z	AD_VP	99999		100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	2001-01-13T00:00:00Z	AD_VP	99999		100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	2000-01-03T00:00:00Z	IT_PROG	9999		102	60
104	Bruce	Ernst	BERNST	590.423.4568	2007-09-21T00:00:00Z	IT_PROG	9999		103	60
105	David	Austin	DAUSTIN	590.423.4569	2005-06-25T00:00:00Z	IT_PROG	9999		103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	2006-02-05T00:00:00Z	IT_PROG	9999		103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	2007-02-07T00:00:00Z	IT_PROG	9999		103	60

At the bottom of the notebook, there is a status bar indicating 'Took 1 sec. Last updated by ML_HR at May 07 2020, 3:30:55 PM (continued)'.

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.

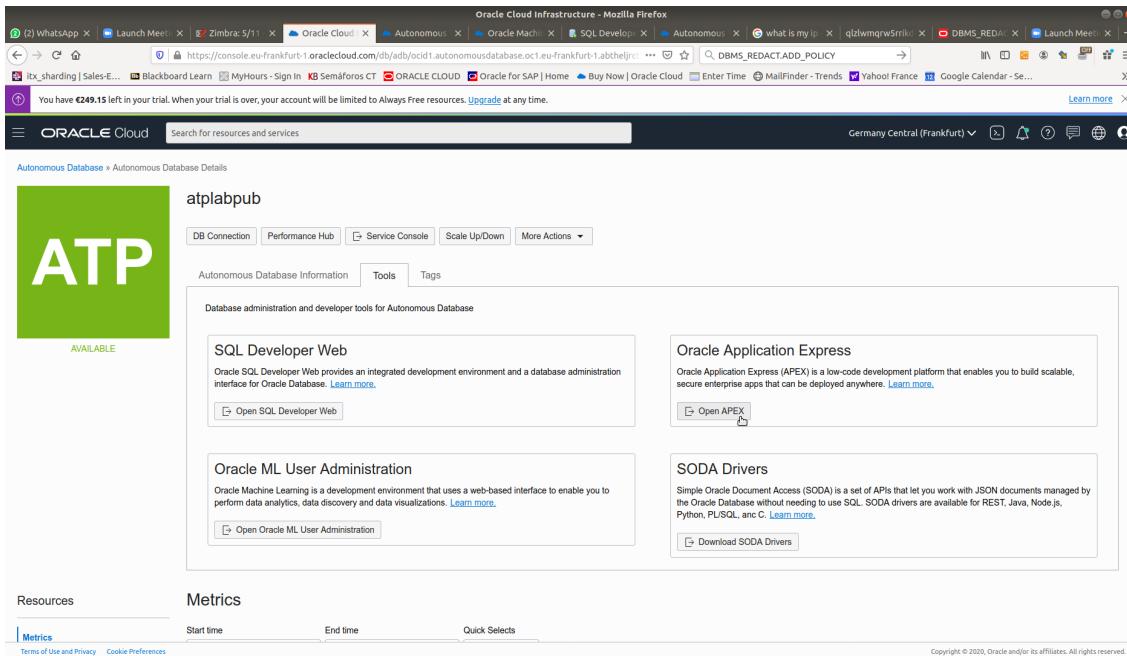
Podemos compararlo con la consulta que se hace con el usuario HR desde SQL Developer web.

The screenshot shows the results of the query 'Select * from hr.employees;' in the Oracle SQL Developer Web interface. The results are displayed in a table with the same columns as the previous screenshot. The data is identical:

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	2003-06-17T00:00:00Z	AD_PRES	99999			90
101	Neena	Kochhar	NKOCHHAR	515.123.4568	2005-09-21T00:00:00Z	AD_VP	99999		100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	2001-01-13T00:00:00Z	AD_VP	99999		100	90
103	Alexander	Hunold	AHUNOLD	590.423.4567	2000-01-03T00:00:00Z	IT_PROG	9999		102	60
104	Bruce	Ernst	BERNST	590.423.4568	2007-09-21T00:00:00Z	IT_PROG	9999		103	60
105	David	Austin	DAUSTIN	590.423.4569	2005-06-25T00:00:00Z	IT_PROG	9999		103	60
106	Valli	Pataballa	VPATABAL	590.423.4560	2006-02-05T00:00:00Z	IT_PROG	9999		103	60
107	Diana	Lorentz	DLORENTZ	590.423.5567	2007-02-07T00:00:00Z	IT_PROG	9999		103	60

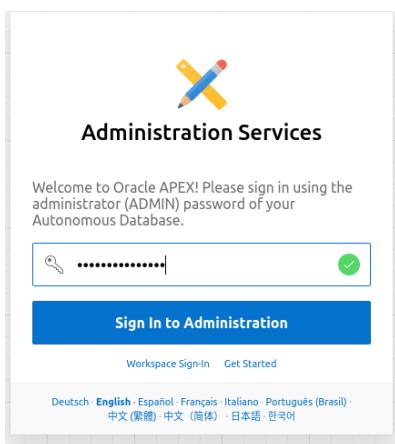
Ejercicio 4: Utilización de APEX

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



The screenshot shows the Oracle Cloud Infrastructure interface. In the top navigation bar, there are several tabs like WhatsApp, Launch Meet!, Zimbra, Oracle Cloud, Autonomous, Oracle Machine, SQL Developer, DBMS_REDACT_ADD_POLICY, and others. Below the navigation bar, the main content area is titled "Autonomous Database Details" for the database "atplabpub". There are four main sections: "Database administration and developer tools for Autonomous Database". The first section is "SQL Developer Web", which provides an integrated development environment and a database administration interface for Oracle Database. The second section is "Oracle Application Express", described as a low-code development platform for building scalable, secure enterprise apps. The third section is "Oracle ML User Administration", which is a development environment for web-based interfaces. The fourth section is "SODA Drivers", which allows working with JSON documents managed by the Oracle Database. At the bottom of the page, there are tabs for "Resources" and "Metrics", and a footer with copyright information.

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



The screenshot shows the "Administration Services" sign-in page for Oracle APEX. It has a logo at the top, followed by a message: "Welcome to Oracle APEX! Please sign in using the administrator (ADMIN) password of your Autonomous Database.". Below this is a password input field with a magnifying glass icon and a green checkmark icon. A large blue button labeled "Sign In to Administration" is centered. At the bottom, there are links for "Workspace Sign-In" and "Get Started", and language options: Deutsch, English, Español, Français, Italiano, Português (Brasil), 中文 (繁體), 中文 (简体), 日本語, and 한국어.

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



The screenshot shows the "Welcome to Oracle Application Express!" page. It features a logo at the top, followed by a message: "Before you get started, please take a moment to create a workspace. A workspace is a shared work area where multiple developers can build applications." Below this, it says: "Once created, sign in to your workspace to begin building applications. Return to Administration Services to create additional workspaces or to manage this Application Express instance." At the bottom, there is a blue button labeled "Create Workspace".



Create Workspace

Identify a new or existing database user to use with your new workspace.

* Database User: HR

* Password: (redacted)

* Workspace Name: WSHR

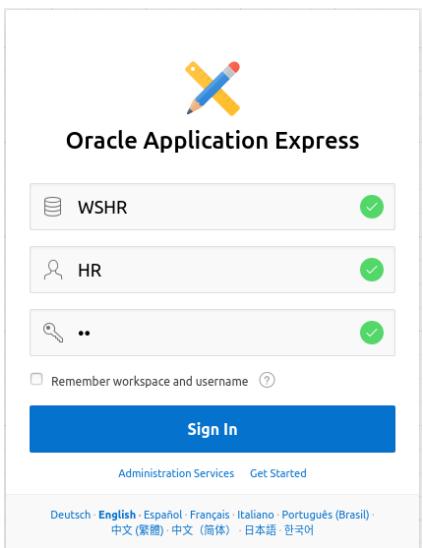
Advanced

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.

The screenshot shows the Oracle Application Express Administration Services interface in Mozilla Firefox. The URL is https://qlzwmqrw5rkkd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/r/p=4550:t=F4550_P1_COMPANY,F4550_P1_USERNAME:WSHR,HR. The main menu includes 'Manage Instance', 'Manage Workspaces' (which is selected), and 'Monitor Activity'. On the right, there's a sidebar with 'About', 'Instance Tasks', 'Feature Configuration', 'Workspace Tasks', and a 'Create Workspace' button. A system message at the bottom left says 'Workspace created. Sign out of Administration Services and sign in to WSHR to begin building applications.' The 'Jobs' section lists several tasks run by the 'ORACLE_APEX...' user, with the last run being 10 minutes ago.

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





Seguimos los pasos siguientes:



Welcome to Oracle Application Express!

Before you get started, please take a moment to set your Application Express (APEX) account password.
Your access to this service is controlled by Single Sign-On (SSO). When your workspace was created, an APEX account was also created with your SSO username and a randomly generated password. Resetting this password is required to run apps you create.
Note: This will not reset your SSO password.

[Set APEX Account Password](#)

Completamos el perfil del usuario HR:

Edit Profile

Profile Details

Workspace	WSHR
Username	HR
* Email Address	pp@gmail.com
First Name	
Last Name	

Profile Photo
Your profile photo personalizes your activity by showing up in the Top Users list. Add, change, or remove your photo.

Photo No file selected.

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”**:



The screenshot shows the Oracle Application Express (APEX) interface. The main navigation bar includes links for Oracle Cloud, Oracle Database, Oracle APEX, App Builder, SQL Workshop, Team Development, App Gallery, and more. The SQL Workshop tab is active. On the left, there's a sidebar with icons for Object Browser, SQL Commands, SQL Scripts, Utilities, and RESTful Services (which is currently selected). The main content area displays 'Top Apps' and 'Top Users' sections. To the right, there's a dashboard with metrics for Applications (0), Tables (8), Productivity Apps (0), and Features (0). The bottom of the page shows a footer with language links (Deutsch, English, Español, Français, Italiano, Português (Brasil)), a URL (https://qlzlwmpqrw5rrkd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/f?p=4850:100:10560686599745::RIR:P0_SELECTED_NODE:OBD), and a reporting timeframe indicator.

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**”:

The screenshot shows the ORDS RESTful Services interface. The top navigation bar includes links for Oracle Cloud, Oracle Database, Oracle APEX, App Builder, SQL Workshop, Team Development, App Gallery, and more. The RESTful Services tab is active. The main content area shows the ORDS Version (19.4.3.r1061746) and a De-Register Schema From ORDS button. It displays three main sections: Schema Access (Access Status: ENABLED), Metadata Access (Authorization Required: ENABLED), and Schema Aliased (Schema Alias: hr). Below these are summary statistics: Total Modules (0), Total Privileges (4), Total Roles (8), and Total Enabled Objects (1). At the bottom, there are sections for Module Status (No Modules Defined), Module Security (No Modules Defined), and Object Aliases (No Aliases Defined).



ORDS Enabled Objects - Mozilla Firefox

RESTful Services \ ORDS RESTful Services \ ORDS Enabled Objects

RESTful Enabled Objects

Parsing Schema	Parsing Object	Object Alias	Type	Status	Auto REST Auth	Ops Allowed	Type Path	Aliased
HR	EMPLOYEES	emp	TABLE	ENABLED	DISABLED	-	ENABLED	(circle)

Legend: (circle) Object name and alias are different (cross) Object name and alias are the same

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

ORDS Enabled Objects - Mozilla Firefox

RESTful Services \ ORDS RESTful Services \ Object Browser

Full Enabled Objects

Parsing Schema	Parsing Object	Object Alias	Type	Status	Auto REST Auth	Ops Allowed	Type Path	Aliased
HR	EMPLOYEES	emp	TABLE	ENABLED	DISABLED	-	ENABLED	(circle)

Legend: (circle) Object name and alias are different (cross) Object name and alias are the same

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



The screenshot shows the Oracle APEX interface with the SQL Workshop tab selected. On the left, there's an Object Browser with a tree view of database objects like COUNTRIES, DEPARTMENTS, JOBS, JOB_HISTORY, LINEORDER, LOCATIONS, and REGIONS. The EMPLOYEES node is selected and highlighted in blue. The main panel displays the structure of the EMPLOYEES table with the following columns:

Column Name	Data Type	Nullable
EMPLOYEE_ID	NUMBER(6,0)	No
FIRST_NAME	VARCHAR2(20)	Yes
LAST_NAME	VARCHAR2(25)	No
EMAIL	VARCHAR2(25)	No
PHONE_NUMBER	VARCHAR2(20)	Yes
HIRE_DATE	DATE	No
JOB_ID	VARCHAR2(10)	No
SALARY	NUMBER(8,2)	Yes
COMMISSION_PCT	NUMBER(2,2)	Yes
MANAGER_ID	NUMBER(6,0)	Yes
DEPARTMENT_ID	NUMBER(4,0)	Yes

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

The screenshot shows the RESTful configuration for the EMPLOYEES table. It includes the following settings:

- REST Enable Object:** Yes
- Object Alias:** emp
- Authorization Required:** No
- RESTful URI:** <https://qlzlwmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/>

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","hire_date":"2003-06-17T00:00:00Z","job_id":"AD_PRES","salary":24000,"commission_pct":null,"manager_id":null,"department_id":90,"links":[{"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/"}]}
```



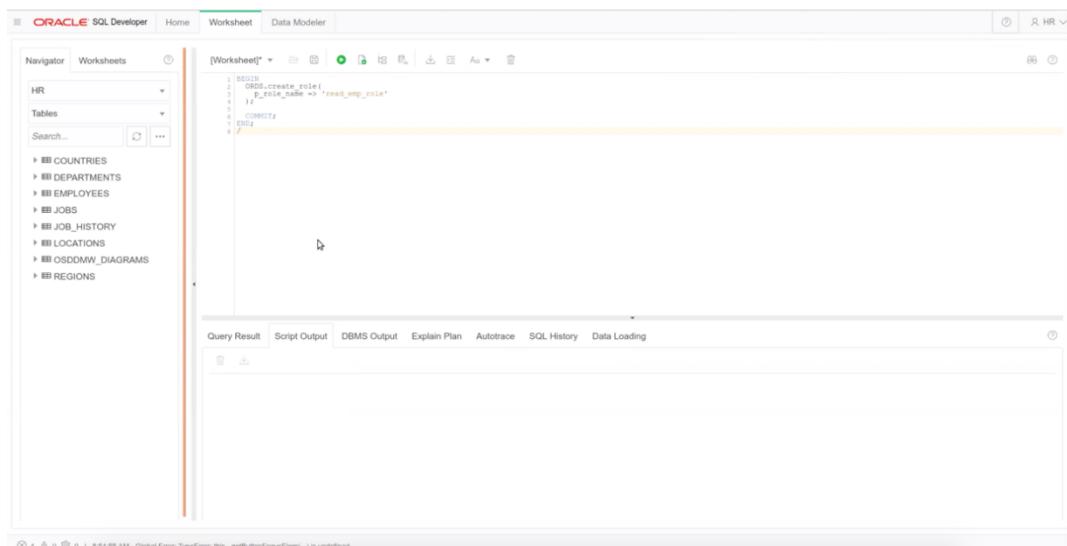
Ejercicio 5 (opcional): Configuración de seguridad de acceso OAuth2

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 “employees” 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;
/
```



The screenshot shows the Oracle SQL Developer interface. The top navigation bar includes 'ORACLE SQL Developer' and 'Home'. Below the navigation bar is a toolbar with icons for file operations like Open, Save, and Print. The left sidebar is titled 'Navigator' and lists 'Workbooks' and 'Worksheets'. Under 'Workbooks', there is an 'HR' entry. Under 'Tables', there is a 'Search...' field and a list of tables: COUNTRIES, DEPARTMENTS, EMPLOYEES, JOBS, JOB_HISTORY, LOCATIONS, OSDMMW_DIAGRAMS, and REGIONS. The main workspace is titled '[Worksheet]*' and contains the following PL/SQL code:

```
1 BEGIN
2   GRDS.create_role(
3     p_role_name => 'read_emp_role'
4   );
5   COMMIT;
6 END;
7 /
8
9 DECLARE
10   l_arr OWA.VC.ARR;
11   l_arr1 := new OWA.VC.ARR;
12   l_arr1(1) := 'read_emp_role';
13   l_arr := l_arr1;
14
15   GRDS.define_privilege (
16     p_privilege_name => 'read_emp_priv',
17     p_label           => 'Employee reader privilege',
18     p_description     => 'Allow to query employees',
19     p_arr             => l_arr
20   );
21
22   COMMIT;
23 END;
24 /
25
```

Below the code, there are tabs for 'Query Result', 'Script Output', 'DBMS Output', 'Explain Plan', 'Autotrace', 'SQL History', and 'Data Loading'. The status bar at the bottom shows the date and time as '8/24/15 10:54 AM - Global Error_TypeError: this._getButtonFocusElement_ is undefined'.

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';
```

The screenshot shows the Oracle SQL Developer interface. The top navigation bar includes tabs for Home, Worksheets, and Data Modeler. On the left, the Navigator pane shows the schema 'HR' and lists various tables: COUNTRIES, DEPARTMENTS, EMPLOYEES, JOBS, JOB_HISTORY, LOCATIONS, OSDDMW_DIAGRAMS, and REGIONS. The Worksheets tab is active, displaying a PL/SQL script to create a role and grant privileges. Below the script, the Query Result tab is selected, showing the results of the executed query. The bottom status bar indicates a global error: 'Global Error TypeError: this._getButtonFocusEvent' is undefined.

```
[Worksheet]*  
1 BEGIN  
2   DBMS_CREATE_ROLE.  
3   P_ROLE_NAME => 'read_emp_role'  
4   P_COMMENT => 'Read Employee Role'  
5 END;  
6 /  
7  
8 DECLARE  
9   l_arr CMA_vc_arri;  
10  l_arr(1) := 'read_emp_role';  
11  
12  DBMS_DEFINE_PRIVILEGE(  
13    P_PRIVILEGE_NAME => 'read_emp_priv',  
14    P_PRIVILEGE_TYPE => 'arr',  
15    P_LABEL => 'Employee reader privilege',  
16    P_DESCRIPTION => 'Allow to query employees',  
17    P_PRIVILEGE_ID => 1);  
18  
19  COMMIT;  
20  
21  
22  SELECT id, name  
23  FROM user_ords_privileges  
24  WHERE name = 'read_emp_priv';  
25  
26  SELECT privilege_id, privilege_name, role_id, role_name  
27  FROM user_ords_privileges  
28  WHERE role_name = 'read_emp_role';
```

Query Result Script Output DBMS Output Explain Plan Autotrace SQL History Data Loading

4 △ 0 ⚙ 0 | 8:45 AM - Global Error TypeError: this._getButtonFocusEvent' is undefined

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';

```

The screenshot shows the Oracle SQL Developer interface. At the top, there's a toolbar with various icons. Below it is a code editor window containing the PL/SQL code. The results tab is selected, showing a single row of data from the query:

privilege_id	name	pattern
10061	read_emp_priv	/emp/*

Below the table, it says "Execution time: 0.064 seconds".

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, válido durante una hora desde su última 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 ultima 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.

The screenshot shows an Oracle SQL Developer worksheet with the following code:

```
1 BEGIN
2   OAUTH.create_client(
3     p_name      => 'Employee Client',
4     p_grant_type => 'client_credentials',
5     p_owner      => 'Employees reader',
6     p_description => 'Client app for employees consultation',
7     p_support_email => 'tim@example.com',
8     p_privilege_names => 'read_emp_priv'
9   );
10
11  COMMIT;
12 END;
13 /
14
15 SELECT id, name, client_id, client_secret
16 FROM   user_ords_clients;
```

The result set is displayed in the Query Result tab:

	id	name	client_id	client_secret
1	10064	Employee Client	mVzhIKydBIvdl...	h0R4l0hJE6XV4...

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;
```



```

1 BEGIN
2   OAUTH.create_client(
3     p_name      => 'Employee Client',
4     p_grant_type => 'client_credentials',
5     p_owner      => 'Employees reader',
6     p_description => 'Client app for employees consultation',
7     p_support_email => 'time@example.com',
8     p_privilege_names => 'read_emp_priv'
9   );
10
11  COMMIT;
12 END;
13 /
14
15 SELECT id, name, client_id, client_secret
16 FROM user_ords_clients;
17
18 SELECT name, client_name
19 FROM user_ords_client_privileges;
20
21 BEGIN
22   OAUTH.grant_client_role(
23     p_client_name =>'Employee Client',
24     p_role_name   =>'read_emp_role'
25   );
26
27  COMMIT;
28 END;
29 /
30
31 SELECT client_name, role_name
32 FROM user_ords_client_roles;

```

Query Result

	client_name	role_name
1	Employee Client	read_emp_role

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://z36w1zyac6l8lpz-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/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
```

