

# HOL5 - ATP tooling

## INDICE

<b>HOL5 - ATP TOOLING .....</b>	<b>1</b>
<b>EJERCICIO 1: CREAR USUARIO DE MACHINE LEARNING.....</b>	<b>3</b>
<b>EJERCICIO 2: UTILIZAR SQL*DEVELOPER WEB.....</b>	<b>5</b>
<b>EJERCICIO 3: CREAR UN NOTEBOOK EN ORACLE MACHINE LEARNING.....</b>	<b>17</b>
<b>EJERCICIO 4: UTILIZACIÓN DE APEX.....</b>	<b>21</b>
<b>EJERCICIO 5 (OPCIONAL): CONFIGURACIÓN DE SEGURIDAD DE ACCESO OAUTH2.....</b>	<b>27</b>



# Ejercicio 1: Crear usuario de Machine Learning

En este ejercicio se explica como 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 a database named 'atplabpub'. At the top, there are several navigation tabs: DB Connection, Performance Hub, Service Console, Scale Up/Down, and More Actions. Below these, the 'Autonomous Database Information' section has tabs for Tools (which is highlighted with a red box) and Tags. Under the Tools tab, there are sections for Database Actions, Oracle Application Express, Oracle ML User Administration, and SODA Drivers. The 'Oracle ML User Administration' section contains a link labeled 'Open Oracle ML User Administration' which is also highlighted with a red box.

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 area 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	Role	Email	Created On	Status
ADMIN	System Administrator	pp@gmail.com	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

Contraseña: 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	Role	Email	Created On	Status
ADMIN	System Administrator	pp@gmail.com	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 Database Actions”.

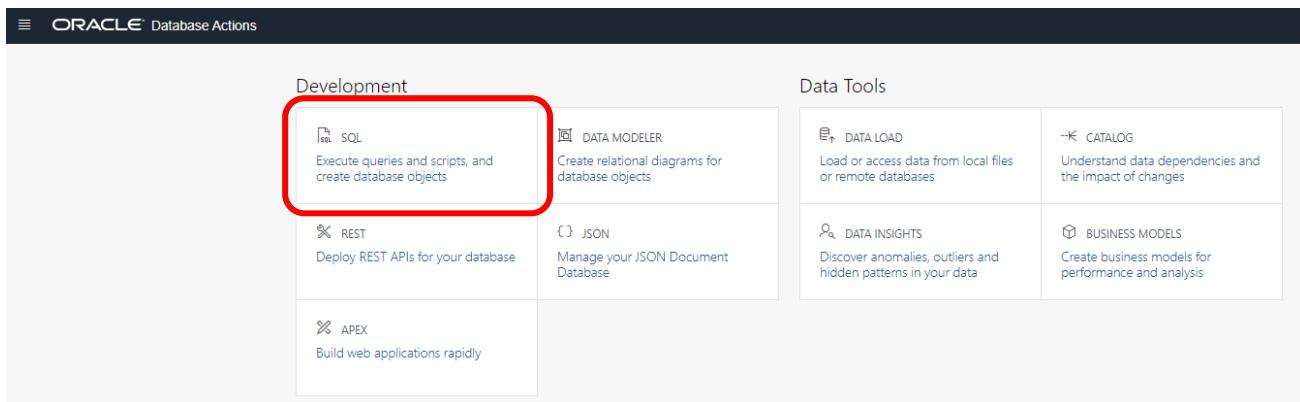
The screenshot shows the 'Autonomous Database Details' page for an instance named 'atplabpub'. At the top, there are several navigation tabs: 'DB Connection', 'Performance Hub', 'Service Console', 'Scale Up/Down', and 'More Actions'. Below these, the 'Tools' tab is selected, indicated by a red box around its border. Under the 'Tools' tab, there are four main sections: 'Database Actions', 'Oracle Application Express', 'Oracle ML User Administration', and 'SODA Drivers'. Each section contains a brief description and a 'View Details' button. The 'Database Actions' section's 'View Details' button is also highlighted with a red box.

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

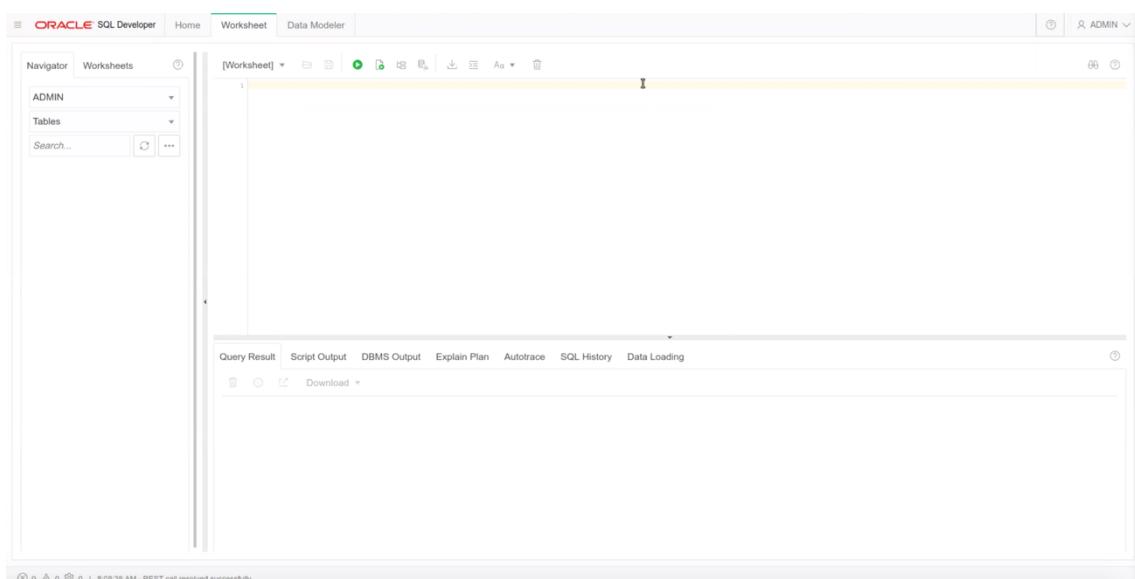
The screenshot shows the 'SQL Developer Web' login page. It features the Oracle logo and the text 'SQL Developer Web'. Below this is a form with two input fields: 'Username' containing 'ADMIN' and 'Password' (the field is empty). At the bottom of the form is a large blue 'Sign in' button.

Ahora vemos la pantalla de **Database Actions**, hacemos click en **SQL**.





Una vez hecho esto accedemos a la pantalla de **SQL Developer Web**, que nos permitirá 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$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;
/
```



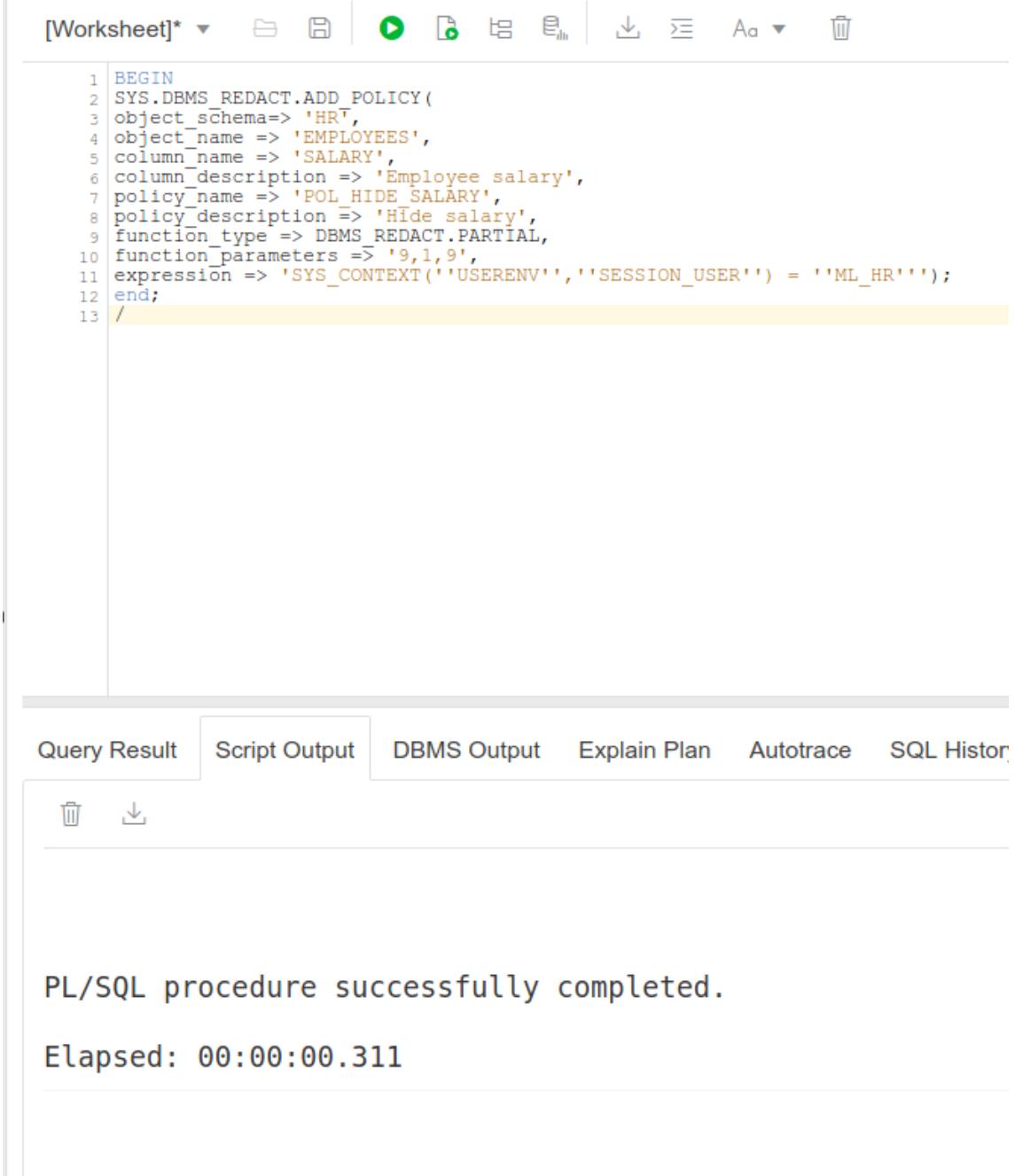
[Worksheet]\* ▾ Aa ▾

```
1 BEGIN
2     ord$admin.enable_schema (
3         p_enabled          => TRUE,
4         p_schema            => 'HR',
5         p_url_mapping_type => 'BASE_PATH',
6         p_url_mapping_pattern => 'hr', -- this flag says, use 'myownsh' in the URIs for MYOWNSH
7         p_auto_rest_auth    => TRUE -- this flag says, don't expose my REST APIs
8     );
9     COMMIT;
10 END;
11 /
12
```

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;
/
```





The screenshot shows a SQL developer interface with a script editor and execution results.

**Script Editor (Query Result tab selected):**

```
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 /
```

**Execution Results:**

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 navigation bar, 'Worksheet' is selected. On the right, a user menu shows 'ADMIN' and a 'Sign Out' option. The main area displays a PL/SQL script in the 'Worksheet' tab:

```

1 BEGIN
2   ORDS_ADMIN.enable_schema (
3     p_schema           => TRUE,
4     p_schemas          => 'HR',
5     p_url_mapping_type => 'STATIC_PATH',
6     p_url_mapping_param  => '',
7     p_auto_index_with  => TRUE
8   );
9 END;
10/

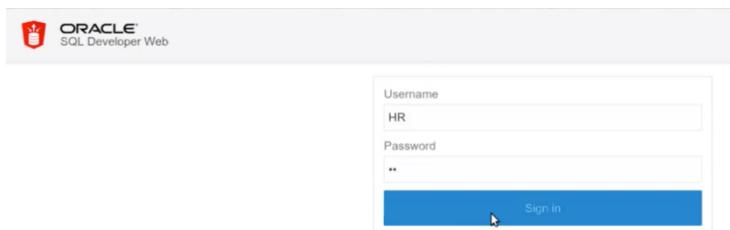
```

Below the code, the status message 'PL/SQL procedure successfully completed.' is displayed, along with an elapsed time of 'Elapsed: 00:00:00.409'.

Luego desde la pagina principal del ATP, pestana “**Tools**”, volvemos a cliquar “**Open Database Actions**”, para volver a la pantalla de login. En la URL, **cambiamos “admin” por “hr”**:

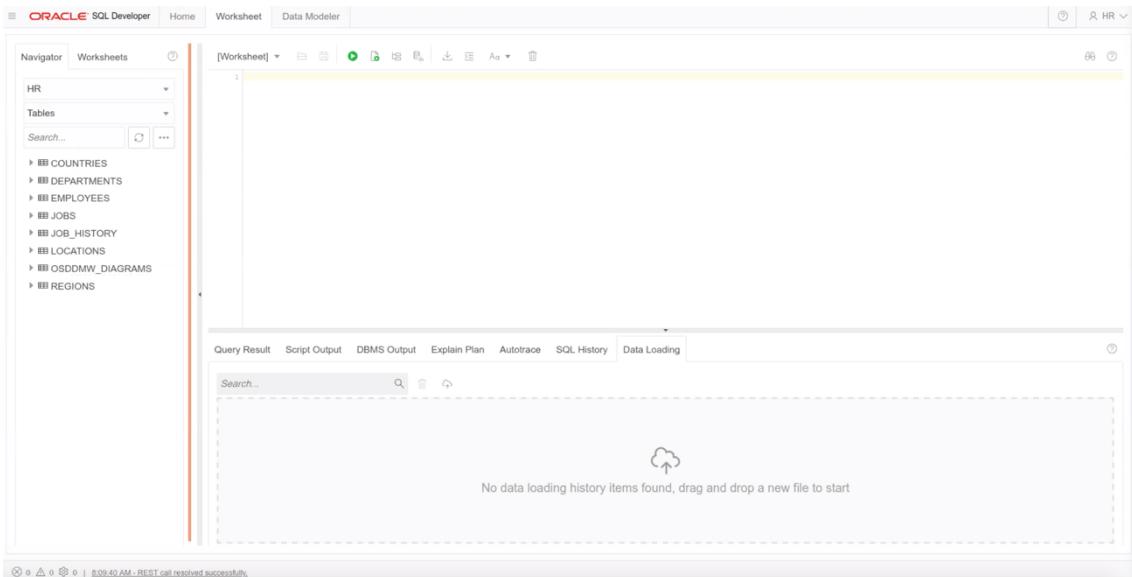


Volverá a aparecer la consola de login, volvemos a introducir el nombre de usuario y contraseña. En este caso el usuario HR/hr o hr/hr (**contraseña siempre en minúsculas**).



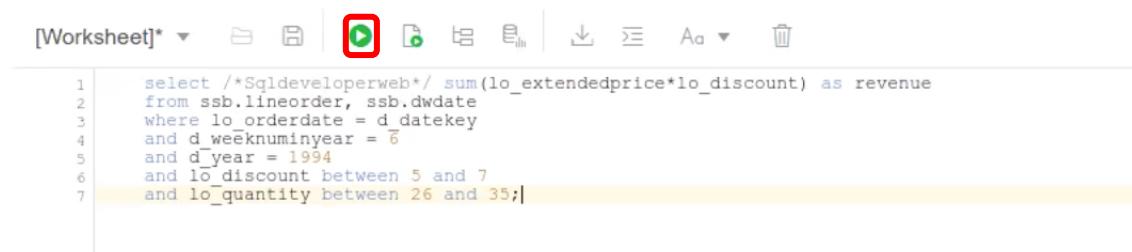
En la siguiente pantalla hacemos click en “**SQL**” para acceder a la misma consola de SQL Developer Web, pero en este caso a la izquierda podemos ver las tablas del esquema HR:





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:

This screenshot shows the Oracle ATP Performance Hub. The top navigation bar includes 'DB Connection' (highlighted with a red box), 'Performance Hub' (also highlighted with a red box), 'Service Console', 'Scale Up/Down', and 'More Actions'. The main content area displays 'Autonomous Database Information' with tabs for 'Tools' and 'Tags'. Below this, there are two sections: 'General Information' (Database Name: atplabpub, Workload Type: Transaction Processing) and 'Infrastructure' (Dedicated Infrastructure: No). A 'Backup' section is also visible.



Una vez aquí, se puede ver la query ejecutándose:

ASH Analytics SQL Monitoring

Top 100 by Last Active Time ▾ Filter by Status, SQL ID or User Name

Status	Duration	Inst ID	SQL ID	SQL Plan Hash	User Name	Parallel	Database Time	I/O Requests	SQL Text
Running	20.00s	1	228ab9f6e2	817007416	HR@Z36W1ZYAC0L8LPZ_ATPLABPUB	1	20.40s	71K	select

Kill Session

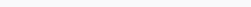
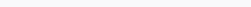
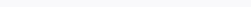
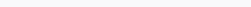
Se puede seleccionar el SQL ID de la query, y acceder a sus detalles

← Back

## Real-time SQL Monitoring

SQL ID: 226q0dh6lmrz Execution ID: 16777216 Status: Executing

Overview

General	Time & Wait	I/O
<p>SQL Text: <code>select * from ( select o_*, row_number() over (order by ...)</code></p> <p>Execution Started: Apr 7, 2020 8:10:08 AM Last Refresh Time: Apr 7, 2020 8:10:36 AM Execution Id: 16777216 User Name: HR@Z36W1ZYAC6L8LPZ_ATPLAB... Fetch Calls: 0</p>	<p>Duration:  28.0s</p> <p>Database:  28.6s</p> <p>Time:  28.6s</p> <p>PL/SQL &amp; Java: 0s</p> <p>Activity:  100%</p>	<p>Buffer:  13M</p> <p>Gets:  99K</p> <p>Lens:  99K</p> <p>Requests:  96.3GB</p> <p>IO:  96.3GB</p> <p>Bytes:  96.3GB</p> <p>Cell Offload Efficiency: 100%</p>

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

## Details

Plan Statistics SQL Text Activity Metrics

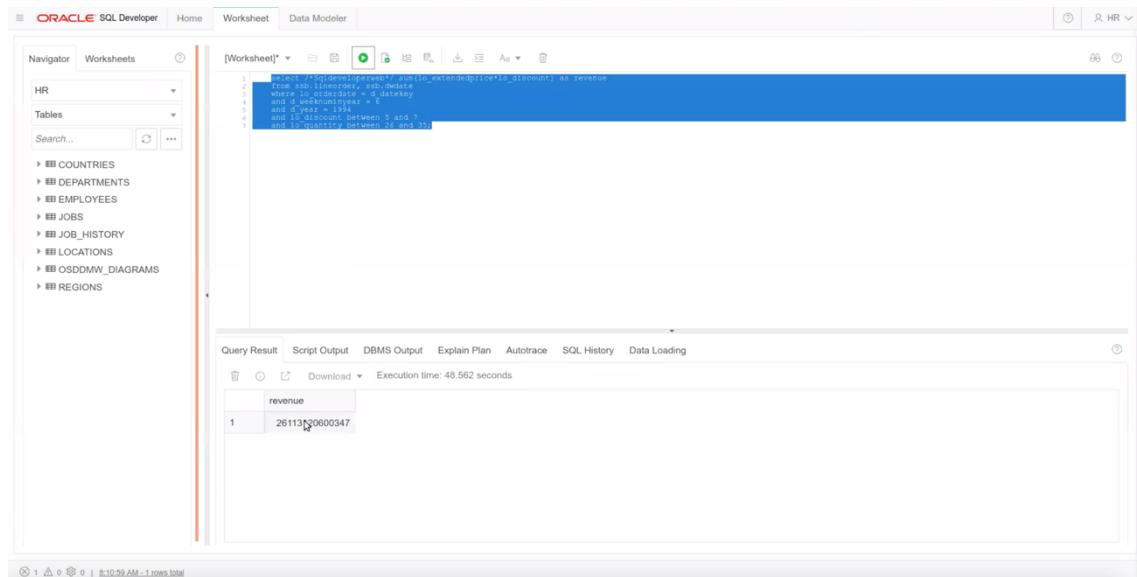
### SQL Binds

#### SQL Text

```
select * from (
    select Q1_qty, row_number() over order by i1_RN___ from
    select Q1_qty, i1_orderkey, sum(i1_extendedprice*i1_discount) as revenue
    from sashlinesales, sub_order
    where sashlinesales.i1_orderkey
    and sashlinesales.i1_orderkey
    and d.warehousekey = 6
    and i1_orderdate between '1997-01-01' and '1997-04-30'
    and i1_discount between 5 and 7
    and i1_quantity between 26 and 35
) Q
where RN___ between :1 and :2
```

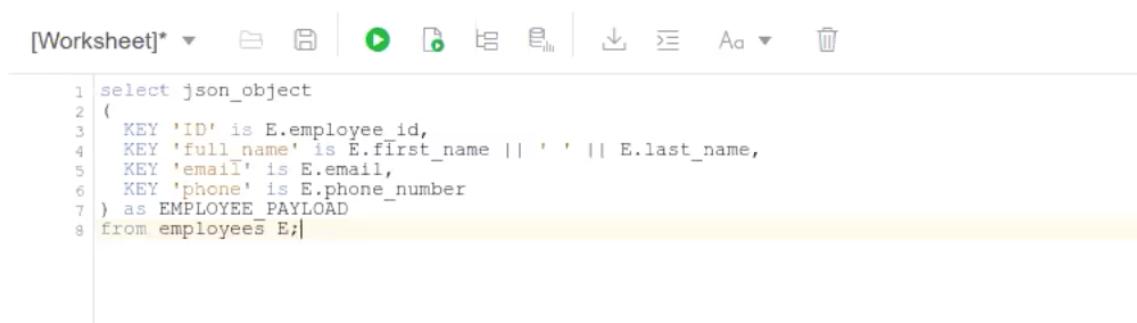


Volviendo a SQL Developer Web, cuando la query haya terminado, podemos ver el resultado de la consulta:

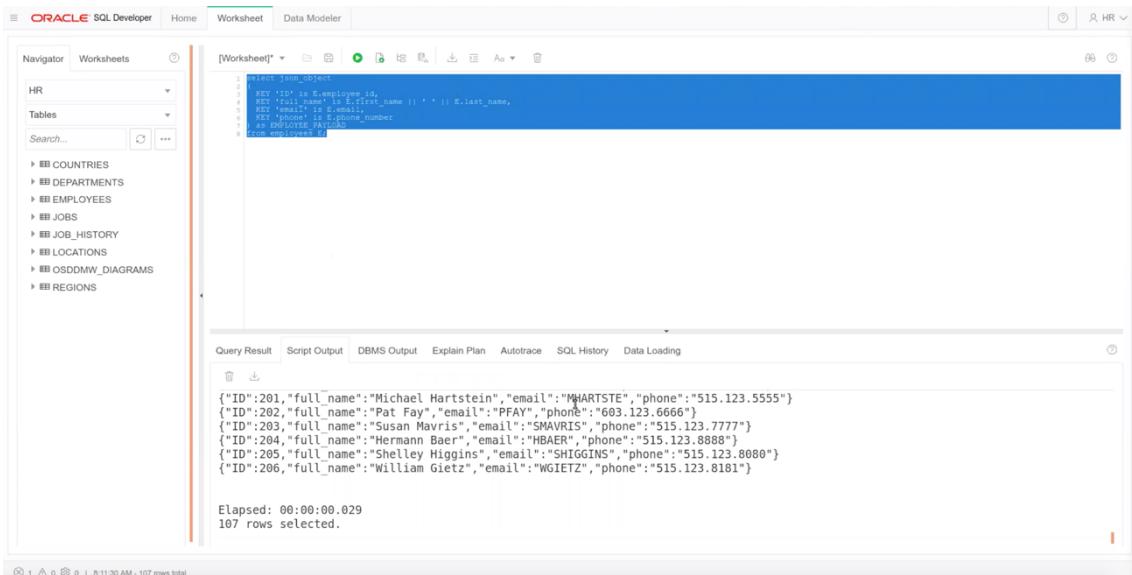


A continuación, ejecutamos una segunda consulta, en este caso la consulta devuelve un **objeto JSON** a partir de los datos de la consulta SQL:

```
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:



The screenshot shows the Oracle SQL Developer interface. In the top navigation bar, 'Worksheet' is selected. The left sidebar shows the 'HR' schema with tables like COUNTRIES, DEPARTMENTS, EMPLOYEES, JOBS, JOB\_HISTORY, LOCATIONS, OSDDMW\_DIAGRAMS, and REGIONS. The main workspace contains a query window with the following code:

```
SELECT * FROM employees
  WHERE first_name = 'Michael'
    AND last_name = 'Hartstein';
```

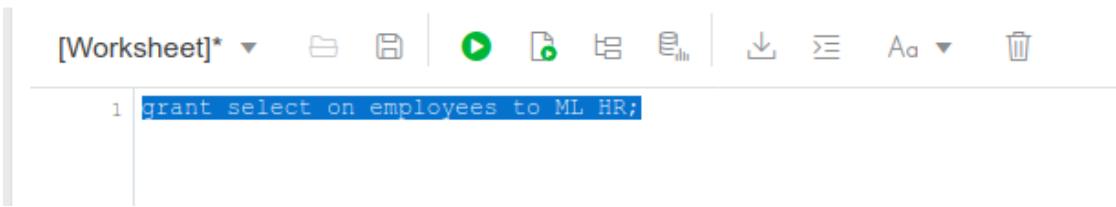
Below the code, the 'Query Result' tab is active, displaying the following JSON output:

```
[{"ID":201,"full_name":"Michael Hartstein","email":"MHAERTSTE","phone":"515.123.5555"}, {"ID":202,"full_name":"Pat Fay","email":"PFAY","phone":"603.123.6666"}, {"ID":203,"full_name":"Susan Mavris","email":"SMAVRIS","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":"515.123.8680"}, {"ID":206,"full_name":"William Gietz","email":"WGIETZ","phone":"515.123.8181"}]
```

At the bottom of the results, it says 'Elapsed: 00:00:00.029' and '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;
```



The screenshot shows the Oracle SQL Developer interface with the 'Worksheet' tab selected. The code 'grant select on employees to ML\_HR;' is entered in the worksheet area. The code is highlighted in blue.

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
11   commit;
12 END;
13 /

```

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”:

Autonomous Database » Autonomous Database Details

**atplabpub**

DB Connection Performance Hub **Service Console** Scale Up/Down More Actions ▾

Autonomous Database Information Tools Tags

Database administration and developer tools for Autonomous Database

**SQL Developer Web**  
Oracle SQL Developer Web provides an integrated development environment and a database administration interface for Oracle Database. [Learn more.](#)

[Open SQL Developer Web](#)

**Oracle ML User Administration**  
Oracle Machine Learning is a development environment that uses a web-based interface to enable you to perform data analytics, data discovery and data visualizations. [Learn more.](#)

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

Autonomous Transaction Processing

Download Oracle Instant Client  
This is a free, light-weight set of tools, libraries and SDKs for building and connecting applications. These libraries underly the Oracle APIs of languages including Node.js, Python and PHP and provide access for OCI, OCCI, JDBC, ODBC and Pro\*C applications. Tools such as SQLPlus and Oracle Data Pump are also included - Oracle recommends using this version of Data Pump for moving existing Oracle Database schemas to Autonomous Transaction Processing.

Download SODA Drivers  
Simple Oracle Document Access (SODA) is a set of APIs for using collections of JSON documents stored in Oracle Database. SODA drivers are available for Java, Node.js, Python, C, PL/SQL, and REST.

Oracle APEX  
Oracle APEX is a low code application development framework for building and deploying world-class data centric applications. APEX provides an easy-to-use browser-based environment to load data, manage database objects, develop REST interfaces, and build applications which look and run great on both desktop and mobile devices.

SQL Developer Web  
Oracle SQL Developer Web provides a browser-based integrated development environment and administration interface for Oracle Autonomous Database. It provides a subset of the features available in the desktop product.

Oracle Machine Learning Notebooks  
Oracle Machine Learning (ML) Notebooks are a collaborative, Apache Zeppelin-based user interface for data scientists and business SQL users of Autonomous Database. ML Notebooks provide an easy-to-use, in-database parallel, distributed machine learning algorithms, in addition to statistical and analytical SQL and PL/SQL functions. ML Notebooks enable sharing of notebooks and templates across the enterprise through permissions-based access, versioning, and execution scheduling.

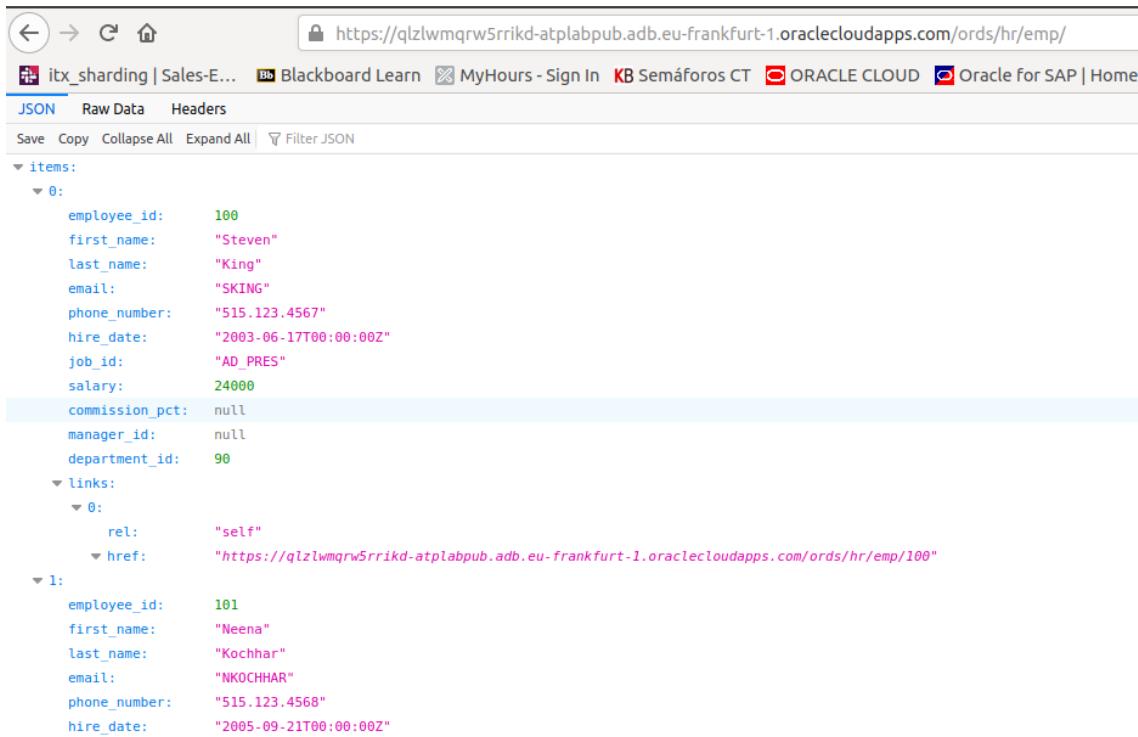
**RESTful Services and SODA**  
Oracle REST Data Services (ORDS) provides HTTPS interfaces for working with the contents of your Oracle Database in one or more REST enabled schemas.

<https://QLZLWMQRW5RRIKD-ATPLABPUB.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/> [Copy URL](#)

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/>



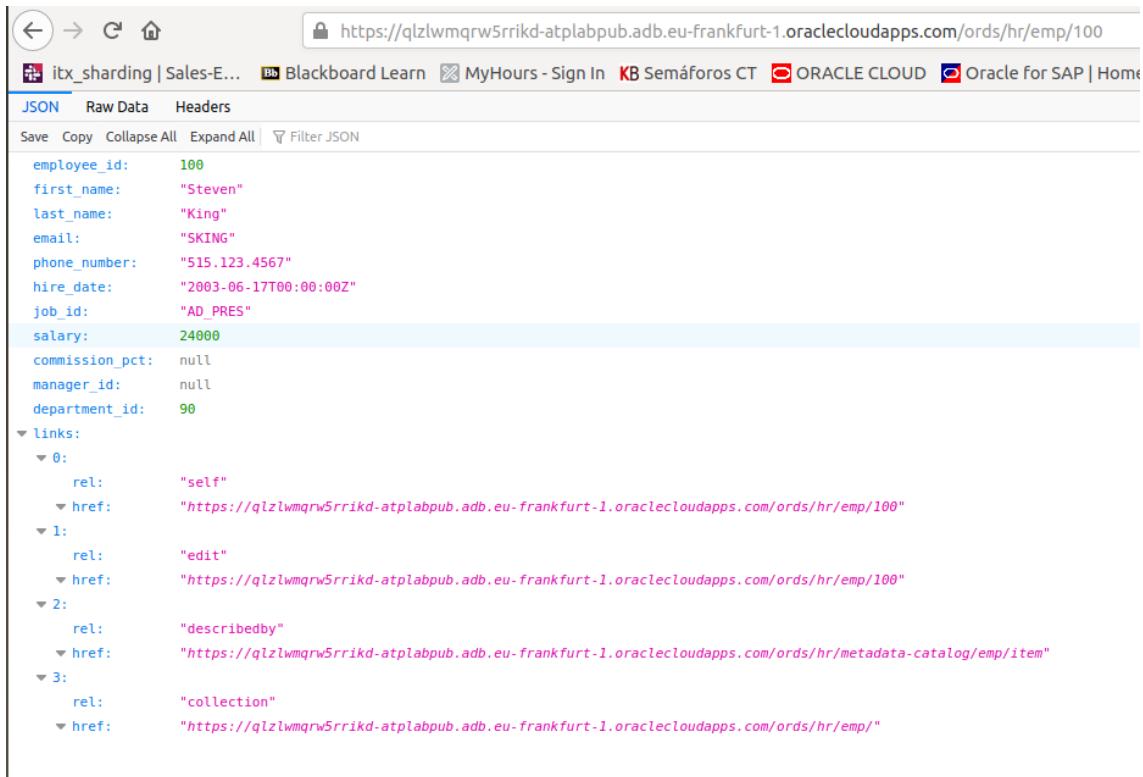
Si pegamos esta URL en un navegador Web, vemos los datos de la tabla “employees”:



The screenshot shows a browser window with the URL <https://qlzlwmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/>. The page displays a JSON response with two items. Item 0 has 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, and department\_id 90. It also includes a self-link. Item 1 has employee\_id 101, first\_name "Neena", last\_name "Kochhar", email "NKOCHHAR", phone\_number "515.123.4568", hire\_date "2005-09-21T00:00:00Z", and other fields like job\_id "AD\_ASST", salary 12000, commission\_pct null, manager\_id 100, and department\_id 90. It includes edit, describedby, and collection links.

```
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”:



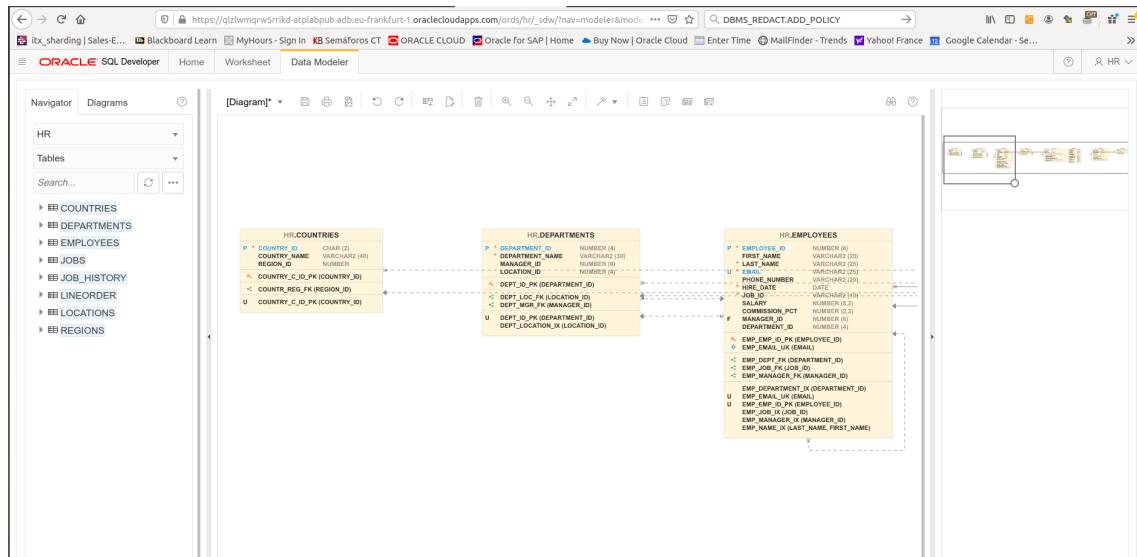
The screenshot shows a browser window with the URL <https://qlzlwmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/100>. The page displays a JSON response for employee ID 100. It includes the same details as item 0 in the previous screenshot, such as first\_name "Steven", last\_name "King", and salary 24000. It also includes a self-link and an edit-link.

```
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 icono de las tres rayas paralelas o hamburger y seleccionamos 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', 'ATPLABPUB' is listed. In the main area, under 'Development', there are several options: 'Download Oracle Instant Client', 'Oracle APEX', 'SQL Developer Web', 'Oracle Machine Learning Notebooks', and 'RESTful Services and SODA'. The 'Oracle Machine Learning Notebooks' option is highlighted with a red box.

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 'Database name:' and 'ATPLABPUB' is entered. Below it, it says 'Sign in with your Oracle Machine Learning Database User credentials'. There are fields for 'USERNAME \*' (containing 'ML\_HR') and 'PASSWORD \*' (containing 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	91dd94jw9tqg	1992170205	HR_ML@Z36W1ZYAC6L8LPZ_ATPLABPUB	2	37.09s	92K	selected
Running	48.00s	1	91dd94jw9tqg	3002741515	HR_ML@Z36W1ZYAC6L8LPZ_ATPLABPUB		48.38s	165K	selected

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

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	<a href="#">91dd94jw9tqg</a>	1992170205	HR_ML@Z36W1ZYAC6L8LPZ_ATPLABPUB	2	37.09s	92K	selected
Running	48.00s	1	<a href="#">91dd94jw9tqg</a>	3002741515	HR_ML@Z36W1ZYAC6L8LPZ_ATPLABPUB		48.38s	165K	selected

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 svb_lineorder, svb_order
        where lo_orderdate = d_datekey
          and l_shipdate between d_syear + 1 and d_syear + 2
          and l_discount between 5 and 7
          and l_quantity between 26 and 35
```

Execution Plan: 2

Execution Started: Apr 7, 2020 10:32:19 AM

Last Refresh Time: Apr 7, 2020 10:32:46 AM

Execution Id: 16777220

User Name: HR\_ML@Z36W1ZYAC6L8LPZ\_ATP...

Fetch Calls: 0

Time & Wait

Duration	Time
28.0s	57.6s

I/O

Buffer Gets	IO Requests	Bytes
19M	147K	142.9GB

Metrics

Details

Plan Statistics Parallel SQL Text Activity Metrics

```
select /*ML.notebook*/
       sum(l_extendedprice*l_discount) as revenue
     from svb_lineorder, svb_order
    where lo_orderdate = d_datekey
      and l_shipdate between d_syear + 1 and d_syear + 2
      and l_discount between 5 and 7
      and l_quantity between 26 and 35
```



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

The screenshot shows the Oracle Machine Learning Notebook interface. The top bar indicates "Connected" to "HR\_ML Project [HR\_ML Workspace...]" and "default". The main area displays a query result titled "REVENUE" with one row: "26113126600347". Below the result is the SQL query:

```
select /*+ no_parallel */ /*+ use_extended_dml(xe_document) */ revenue
  From xtb_licenses, xtb_dialect
 where xtb_licenses.license_id = xtb_dialect.license_id
   and xtb_licenses.year = 4
   and xtb_licenses.month = 1
   and xtb_dialect.between 5 and 7
   and xtb_dialect.between 26 and 30;
```

At the bottom, there is a "sql" tab and a "READY" status indicator.

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

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

The screenshot shows the Oracle Machine Learning Notebook interface. The main area displays a query result for the "hr.employees" table. The columns are: EMPLOYEE\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, HIRE\_DATE, JOB\_ID, SALARY, COMMISSION\_PCT, MANAGER\_ID, and DEPARTMENT\_ID. The data includes rows for employees like Steven King, Neena Kochhar, etc. The "SALARY" column values are all masked with "9999". Below the result is the SQL query:

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

At the bottom, there is a "sql" tab and a "READY" status indicator.

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 Oracle SQL Developer Web interface. The left sidebar shows the "Navigator" with "Tables" selected, showing "HR" and "EMPLOYEES". The main area shows the query result for "hr.employees" with the following data:

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	2006-01-03T00:00:00Z	IT_PROG	9999		102	60
104	Bruce	Ernst	BERNST	590.423.4568	2007-05-21T00:00:00Z	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 ATP Autonomous Database Details page. On the left, there's a green box labeled "ATP" with "AVAILABLE" below it. At the top, there are several tabs: DB Connection, Performance Hub, Service Console, Scale Up/Down, and More Actions. The "Tools" tab is highlighted with a red box. Below the tabs, there are sections for Autonomous Database Information, Database Actions, Oracle ML User Administration, Graph Studio, Oracle Application Express, and SODA Drivers. The "Oracle Application Express" section is also highlighted with a red box. It contains a brief description of Oracle Application Express (APEX) and a "Open APEX" button.

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

The screenshot shows the Oracle APEX sign-in page for Administration Services. It features a logo of two crossed pencils. The page text says: "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 key icon and a visibility toggle. A large green "Sign In to Administration" button is at the bottom. At the very bottom, there are language links: 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”





## Welcome to Oracle Application Express!

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.

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.

[Create Workspace](#)

### Create Workspace

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

\* Database User: HR

\* Password:

\* Workspace Name: WSHR

[Advanced](#)

[Cancel](#) [Create Workspace](#)

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://qlzlwqmqrw5rrid-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/r/p4050/3.1123693232071>. The page title is "Application Express Administration Services - Mozilla Firefox".

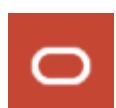
The main navigation bar includes links for Launch Me!, Oracle Cloud, Application, Autonomous, Oracle Machine, SQL Development, Autonomous, DBMS REDACT, and Launch Me!.

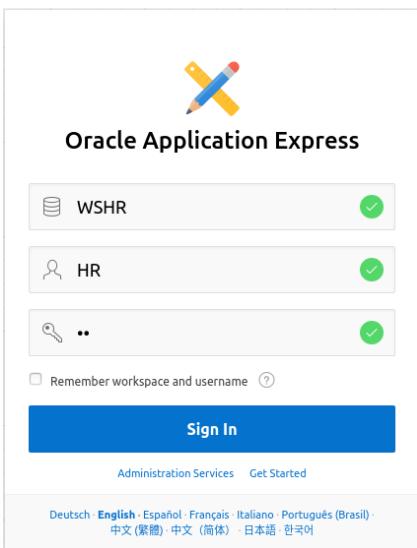
The left sidebar has sections for ORACLE APEX (Manage Instance, Manage Workspaces, Monitor Activity), Instance Administration (Manage Instance, Manage Workspaces, Monitor Activity), System Message (Workspace Summary, Workspaces, Schemas, Applications, Users, Mail Queue Entries, Websheets), and Jobs (ORACLE\_APEX\_AUTO\_APPROVAL, ORACLE\_APEX\_DAILY\_MAINTENANCE, ORACLE\_APEX\_MAIL\_QUEUE, ORACLE\_APEX\_PURGE\_SESSIONS, ORACLE\_APEX\_W5\_NOTIFICATIONS).

The right sidebar provides access to various tasks: About, Instance Tasks (Feature Configuration, Workspace Tasks, Create Workspace, Find a Workspace, Manage Workspaces), Available Updates (System is up-to-date, Once a new Application Express release becomes available, your instance will be immediately upgraded), and a link to Learn More.

A message at the top states: "Workspace created. Sign out of Administration Services and sign in to [WSHR](#) to begin building applications."

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.

[Cancel](#) [Apply Changes](#)

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 in Mozilla Firefox. The URL is [https://qlzwmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/f?p=4500:100:10560686599::DBMS\\_REDACT\\_ADD\\_POLICY](https://qlzwmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/f?p=4500:100:10560686599::DBMS_REDACT_ADD_POLICY). The main menu bar includes links like Oracle Cloud, Autonomous, Oracle Machine Learning, Oracle Development, Autonomous Database, what is my IP, DBMS REDACT, Launch Me!, etc. The left sidebar has sections for ORACLE APEX, App Builder, SQL Workshop, Team Development, and App Gallery. The SQL Workshop section is currently selected. A tooltip for 'RESTful Services' is shown above a button in the sidebar. The main content area displays 'Top Apps', 'Top Users', 'News and Messages', and various statistics for Applications (0), Tables (8), Productivity Apps (0), and Features (0). On the right, there's a sidebar for 'About' (describing Oracle APEX), 'Dashboard' (with 0 applications and 8 tables), 'Resources' (Community Site, Blog, Hands-On Labs, Education, Social), and social media links.

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 in Mozilla Firefox. The URL is [https://qlzwmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/f?p=4850:100:10560686599::DBMS\\_REDACT\\_ADD\\_POLICY](https://qlzwmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/f?p=4850:100:10560686599::DBMS_REDACT_ADD_POLICY). The main menu bar includes links like Oracle Cloud, Autonomous, Oracle Machine Learning, Oracle Development, Autonomous Database, what is my IP, DBMS REDACT, Launch Me!, etc. The left sidebar has sections for RESTful Services, ORDS RESTful Services, Team Development, and App Gallery. The RESTful Services section is currently selected. The main content area displays the ORDS Version (19.4.3.r1061746), Schema Access (Access Status: ENABLED), Metadata Access (Authorization Required: ENABLED), and Schema Aliased (Schema Alias: hr). Below this, it shows a summary of modules (0), privileges (4), roles (8), and total enabled objects (1). The bottom section shows module status, module security, and object aliases.



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	HR	<span style="color: green;">(green circle)</span>

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

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

Object Browser

Parsing Schema	Parsing Object	Object Alias	Type	Status	Auto REST Auth	Ops Allowed	Type Path	Aliased
HR	EMPLOYEES	emp	TABLE	ENABLED	DISABLED	ENABLED	HR	<span style="color: green;">(green circle)</span>

Legend: (green circle) Object name and alias are different (red circle) 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**.

EMPLOYEES		
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**:

REST Enable Object	<input checked="" type="radio"/> Yes	<input type="radio"/> No	(?)
Object Alias	emp <a href="#">(?)</a>		
Authorization Required	<input type="radio"/> Yes	<input checked="" type="radio"/> No	(?)
RESTful URI	<a href="https://qlzlwqmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/">https://qlzlwqmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/</a> <a href="#">(?)</a>		

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://qlzlwqmqrw5rrikd-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://qlzlwqmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/100"}, {"rel":"edit","href":"https://qlzlwqmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/100"}, {"rel":"describedby","href":"https://qlzlwqmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/metadata-catalog/emp/item"}, {"rel":"collection","href":"https://qlzlwqmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/"}]}
```



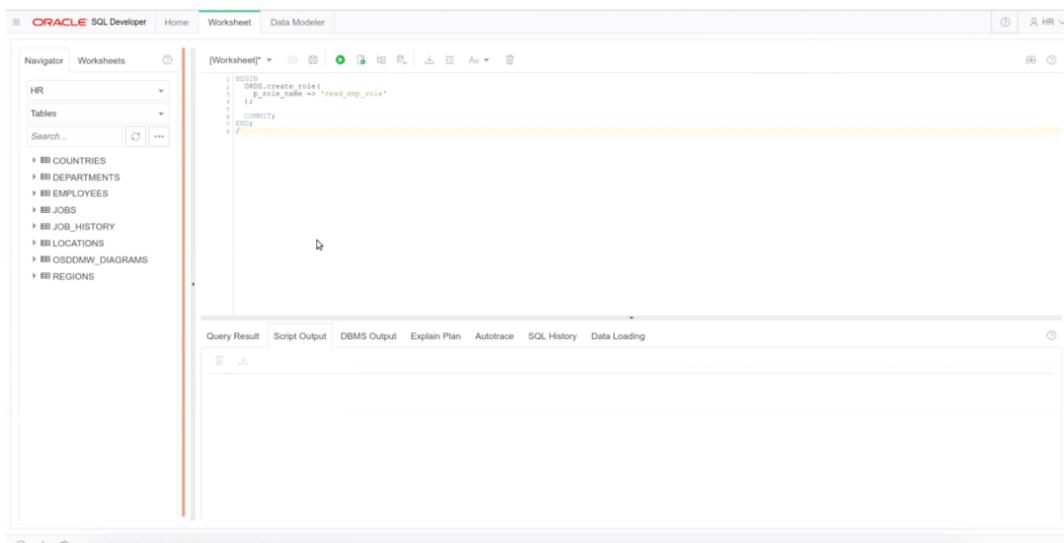
# Ejercicio 5 (opcional): Configuración de seguridad de acceso OAuth2

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', 'Home', 'Worksheet' (which is selected), and 'Data Modeler'. Below the navigation bar is a toolbar with various icons for file operations like Open, Save, Print, and Undo.

The left sidebar contains a 'Navigator' section with 'HR' selected, and a 'Tables' section listing 'COUNTRIES', 'DEPARTMENTS', 'EMPLOYEES', 'JOBS', 'JOB\_HISTORY', 'LOCATIONS', 'OSDMW\_DIAGRAMS', and 'REGIONS'. A 'Search...' field is also present.

The main workspace is titled '[Worksheet]\*' and displays the following PL/SQL code:

```
1 BEGIN
2   OGEDS.create_role(
3     p_role_name => 'read_emp_role'
4   );
5
6   COMMIT;
7 END;
8 /
9
10 DECLARE
11   l_arr OWA_vc_ARR;
12   l_val VARCHAR2(100);
13   l_arr(1) := 'read_emp_role';
14
15   OGEDS.define_privilege (
16     p_privilege_name => 'read_emp_priv',
17     p_label           => 'Employee reader privilege',
18     p_description      => 'Allow to query employees',
19     p_description      => 'Allow to query employees'
20   );
21
22   COMMIT;
23 END;
24 /
25
```

Below the code editor are tabs for 'Query Result', 'Script Output', 'DBMS Output', 'Explain Plan', 'Autotrace', 'SQL History', and 'Data Loading'. The status bar at the bottom shows '4 4 0 0 | 8:54:55 AM - Global Error: TypeError: this.\_definitionFocusElement\_ 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, Worksheet, and Data Modeler. The Worksheet tab is active, displaying a PL/SQL script. The Navigator panel on the left lists various database objects like COUNTRIES, DEPARTMENTS, EMPLOYEES, JOBS, JOB\_HISTORY, LOCATIONS, OSDDMW\_DIAGRAMS, and REGIONS. The main workspace shows the following PL/SQL code:

```
1 BEGIN
2   GRSG.create_role(
3     p_role_name => 'read_emp_role'
4   );
5
6   COMMIT;
7
8   END;
9
10
11  DECLARE
12    l_arr CMA.vc_arr2;
13    l_val 1;
14
15    l_arr(l) := 'read_emp_role';
16
17    GRSG.define_privilege (
18      p_privilege_name => 'read_emp_priv',
19      p_label          => 'Employee reader privilege',
20      p_description    => 'Allow to query employees'
21    );
22
23    COMMIT;
24
25  END;
26
27
28  SELECT id, name
29  FROM user_ords_privileges
30  WHERE name = 'read_emp_priv';
31
32  SELECT privilege_id, privilege_name, role_id, role_name
33  FROM user_ords_privileges
34  WHERE privilege_name = 'read_emp_priv';
35
36  WHERE role_name = 'read_emp_role';
```

Below the code, the Query Result tab is selected, showing a single row of results:

id	name
1	read_emp_priv



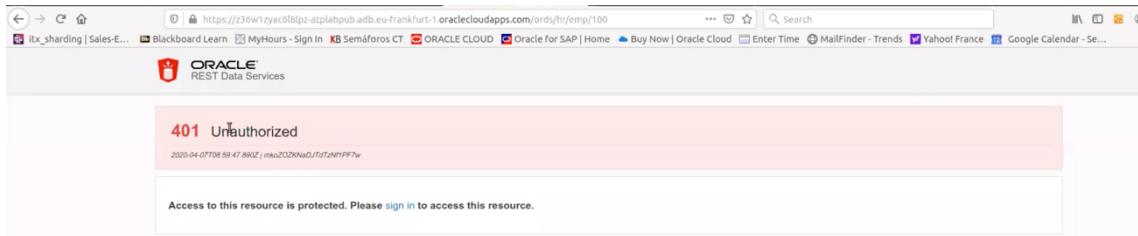
A continuación, mapeamos el privilegio a todas las terminaciones del endpoint “/emp/\*”.

```
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. A PL/SQL block is being run, and its output is displayed in the 'Query Result' tab. The result is a single row:

privilege_id	name	pattern
1	10061	/emp/*

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.

The screenshot shows a SQL 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  COMMIT;
11 END;
12 /
13
14 SELECT id, name, client_id, client_secret
15 FROM user_ords_clients;
```

Below the code, the "Query Result" tab is selected, showing the execution time: 0.034 seconds. The result table contains one row:

	id	name	client_id	client_secret
1	10064	Employee Client	mVzhIKydBlVdL...	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 => 'tim@example.com',
8     p_privilege_names => 'read_emp_priv'
9   );
10  COMMIT;
11 END;
12 /
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://z36w1zyac618lpz-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://z36w1zyac618lpz-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/100
```

