

HOL5 - ATP tooling

INDICE

HOL5 - 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.....	17
EJERCICIO 4: UTILIZACIÓN DE APEX.....	21
EJERCICIO 5 (OPCIONAL): CONFIGURACIÓN DE SEGURIDAD DE ACCESO OAUTH2.....	27



Ejercicio 1: Crear usuario de Machine Learning

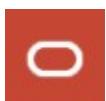
En este ejercicio se explica como crear un usuario de Oracle Machine Learning. Este usuario lo utilizaremos en ejercicios posteriores para crear un Notebook y consultar datos del esquema HR. En la pantalla principal del ATP, elegir la pestaña “Tools”, y luego cliquear el botón “**Open Oracle ML User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

The screenshot shows the Oracle Autonomous Database Details page for a database named 'atplabpub'. At the top, there is a green 'ATP' logo with 'AVAILABLE' status. Below it, the 'Tools' tab is highlighted with a red box. In the 'Oracle ML User Administration' section, the 'Open Oracle ML User Administration' button 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 a 'SIGN IN' page for Oracle Machine Learning. It features a large blue header with a cloud icon. The main form asks for a 'Database name:' which is filled with 'ATPLABPUB'. Below it, it says 'Sign in with your Oracle Machine Learning Database Administrator credentials'. The 'USERNAME *' field contains 'ADMIN' and the 'PASSWORD *' field contains a masked password. The 'Sign In' button at the bottom is highlighted with a red box.

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

Clipear 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		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 Database Actions”.

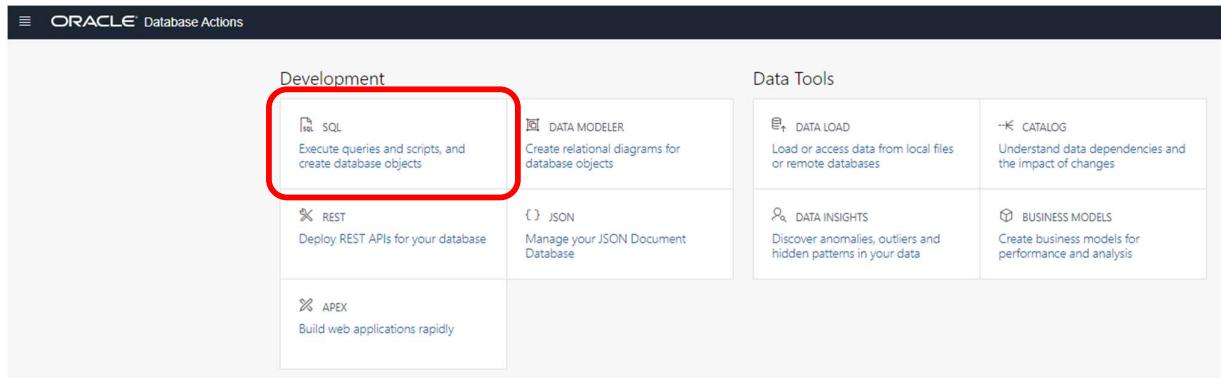
The screenshot shows the Oracle Autonomous Database Details page. At the top, there is a navigation bar with links for Overview, Autonomous Database, and Autonomous Database Details. Below the navigation bar, there is a green banner with the text "ATP" and "AVAILABLE". The main content area has tabs for DB Connection, Performance Hub, Service Console, Scale Up/Down, More Actions, Autonomous Database Information, Tools (which is highlighted with a red box), and Tags. Under the Tools tab, there are several sections: "Database Actions" (with a sub-section "Open Database Actions" highlighted with a red box), "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). The "Autonomous Database Information" section is also visible.

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

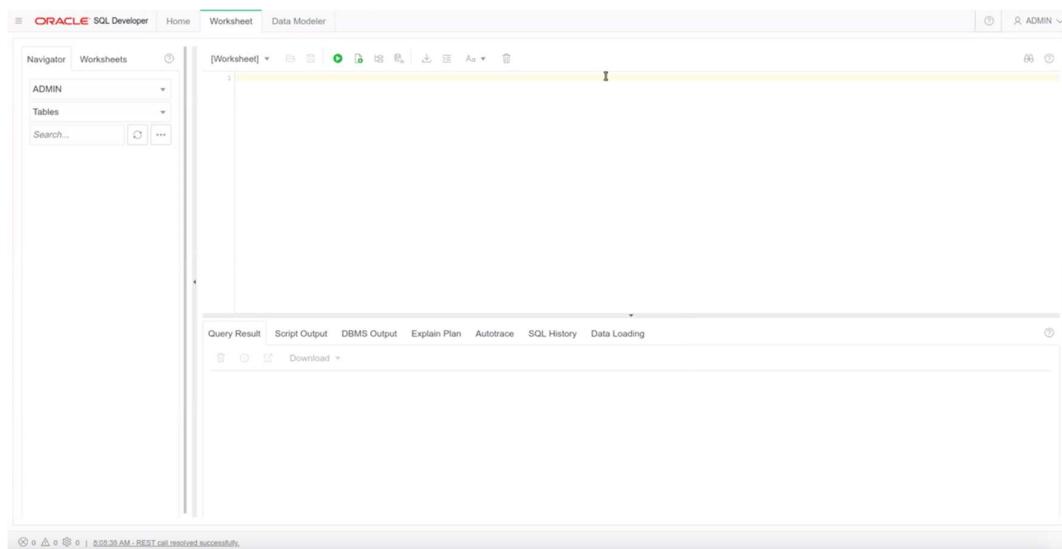
The screenshot shows the Oracle SQL Developer Web login screen. It features the Oracle logo and the text "SQL Developer Web". Below the logo is a form with two input fields: "Username" containing "ADMIN" and "Password" (the field is empty). At the bottom of the form is a large blue "Sign in" button.

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.

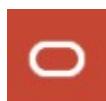


A continuación, ejecutamos el siguiente código, que **habilita el usuario HR en ORDS** y lo habilita para poder acceder con **SQL Developer Web**:

```

BEGIN
    ordbs_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;
/

```





The screenshot shows a Oracle SQL Developer worksheet window. The title bar says "[Worksheet]*". The toolbar includes icons for back, forward, execute, and save. The code area contains the following PL/SQL block:

```
1 BEGIN
2   ords_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;
/
```



[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 History



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.



```

1 BEGIN
2   GRANT ADMIN_ENABLE_SCHEMA (
3     P_SCHEMA          => TRUE,
4     P_ACTIONS         => TRUE,
5     P_MAPPING_TYPE   => 'ROLE_TO_PATH',
6     P_PATH            => '*/ADMIN/*',
7     P_GRANT_PRIVILEGE => TRUE
8   ) TO PUBLIC;
9
10 END;
11 /

```

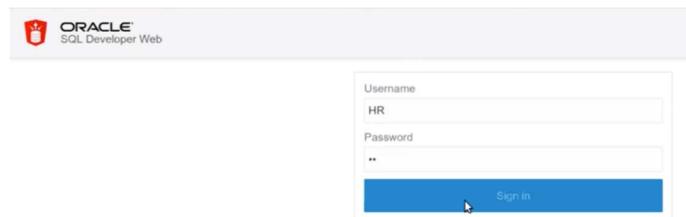
PL/SQL procedure successfully completed.
Elapsed: 00:00:00.409

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

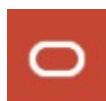
https://z36w1zyac6l8lpz-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/admin/sign-in/?r=_sdw%2F%

https://z36w1zyac6l8lpz-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/sign-in/?r=_sdw%2F%3Fnav%3Dworksheet →

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:



Una vez aquí, se puede ver la query ejecutándose:

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

[← Back](#)

Real-time SQL Monitoring

SQL ID: 228e0d8f6mrz Execution ID: 16777216 Status: Executing

Overview

General

SQL Text: `select * from t select a, * , row_number() over (order by ...)`

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@Z3DW1ZYAC6L8LPZ_ATPLAB...

Fetch Calls: 0

Time & Wait

Duration	Time
28.0s	25.6s

Database: PL/SQL & Java: Os Activity: 100%

I/O

Buffer Gets	I/O Requests	I/O Bytes
13M	99K	96.3GB

Cell Offload Efficiency: 100%

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



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

The screenshot shows the Oracle SQL Developer Web interface. In the top navigation bar, 'ORACLE SQL Developer' is selected. Below it, the 'Worksheet' tab is active. On the left, the 'Navigator' pane shows a tree view of tables under the 'HR' schema, including COUNTRIES, DEPARTMENTS, EMPLOYEES, JOBS, JOB_HISTORY, LOCATIONS, OSDDMW_DIAGRAMS, and REGIONS. The main workspace displays a query in the 'Worksheet' tab:

```
select /*+ replace(reorder, sub-select) */ sum(s.extendedprice*t.discount) as revenue
from products s
join suppliers r
join salespeople p
join regions t
join nation n
join countries c
where r.supplier_id = s.supplier_id
and p.salesperson_id = s.salesperson_id
and t.region_id = s.region_id
and n.nation_id = t.nation_id
and c.country_id = n.country_id
and r.discount between 1 and 10
and t.discount between 10 and 20
```

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

revenue
2611230600347

At the bottom of the interface, there are status indicators: 1 row, 0 errors, 0 warnings, and a total execution time of 48.562 seconds.

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;
```

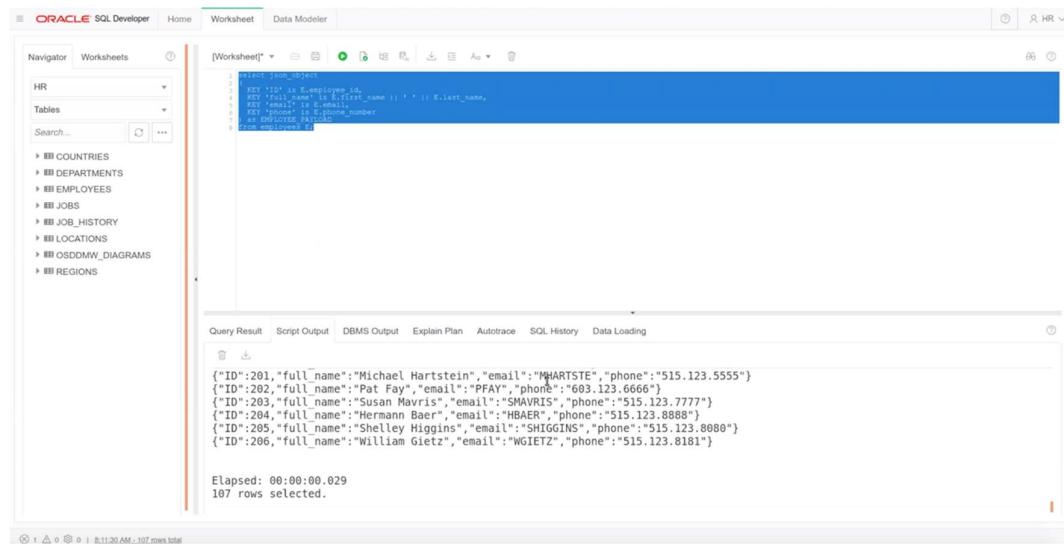
The screenshot shows the Oracle SQL Developer Web interface with the same layout as the previous one. The 'Worksheet' tab is active. The query in the worksheet is identical to the one above:

```
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 'Query Result' tab is not visible in this specific screenshot, but the code is present in the worksheet area.



Podemos ver el resultado en formato JSON:



The screenshot shows the Oracle SQL Developer interface with the HR schema selected. A query is run in the worksheet:

```
select json_object(
    select * from employees
) as employee_json
from dual;
```

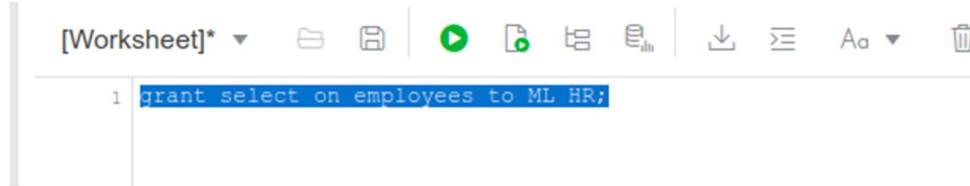
The results are displayed in the Query Result tab as JSON objects:

```
[{"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":"SMAVRIS","phone":"515.123.7777"}, {"ID":204,"full_name":"Herman 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;
```

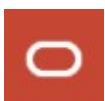


The screenshot shows the Oracle SQL Developer interface with the [Worksheet] tab selected. The following SQL command is entered:

```
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 recuperaremos el REST Endpoint desde la página principal del ATP. Hacer click en el botón “Service Console”:

The screenshot shows the Oracle Autonomous Database Service Console for an instance named "atplabpub". The "Service Console" tab is highlighted with a red box. Below it, there are tabs for "Autonomous Database Information", "Tools", and "Tags". Under "Database Actions", there is a section titled "Database Actions" with a button labeled "Open Database Actions". Another section titled "Oracle ML User Administration" is also visible.

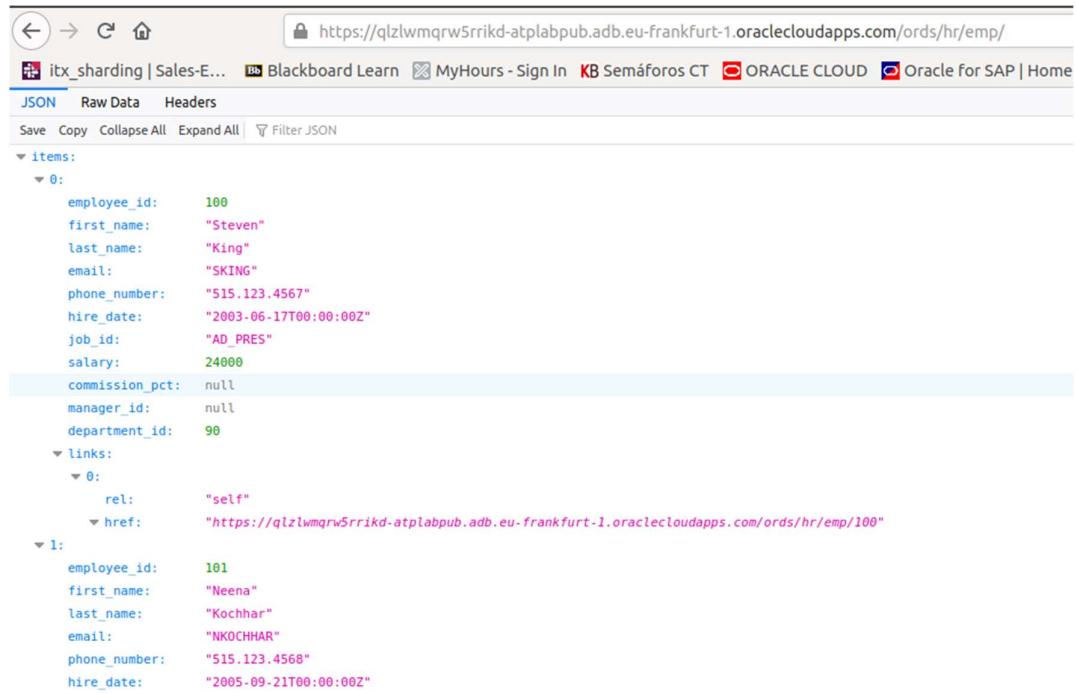
En la pantalla siguiente, hacer click en “Development”. En el apartado “**RESTful Services and SODA**”, vemos nuestro **REST Endpoint**:

The screenshot shows the "Development" section of the Oracle Autonomous Transaction Processing page for database "ATPLABPUB". It lists several services: "Download Oracle Instant Client", "Download SODA Drivers", "Oracle APEX", "SQL Developer Web", "Oracle Machine Learning Notebooks", and "RESTful Services and SODA". The "RESTful Services and SODA" section is highlighted with a red box, showing the URL "https://QLZLWMQRW5RIKD-ATPLABPUB.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/" with a "Copy URL" button.

Para consultar nuestra tabla, a la URL de REST Endpoint le añadimos “/hr/emp/”, por ejemplo:
<https://QLZLWMQRW5RIKD-ATPLABPUB.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/>



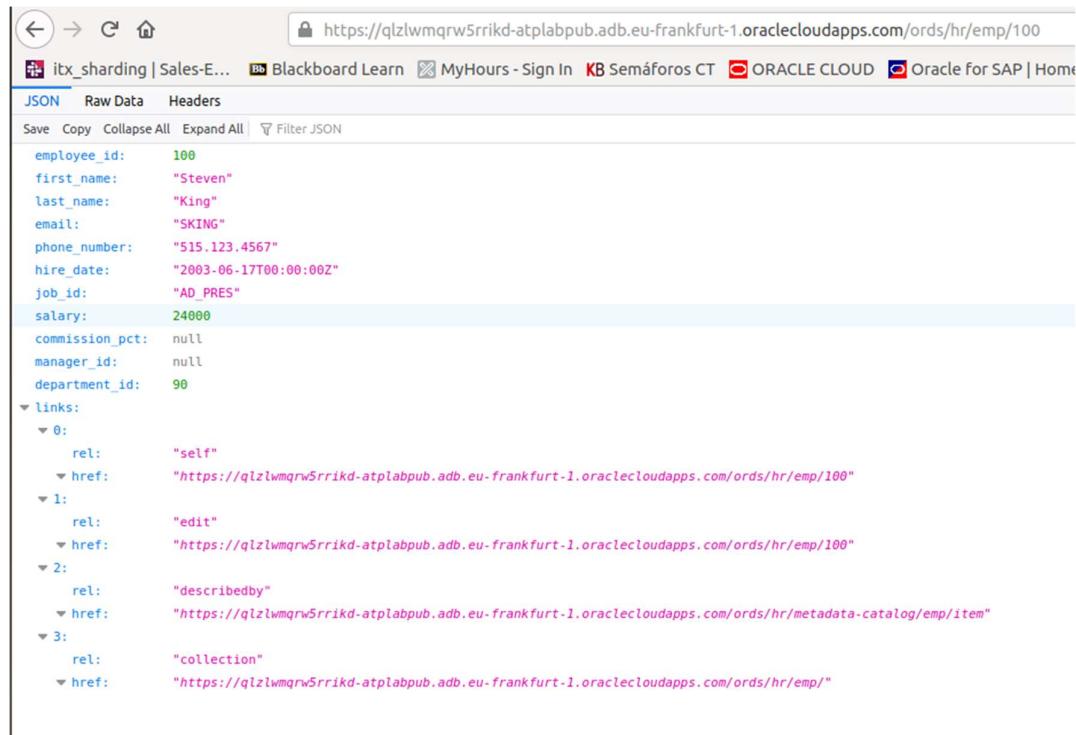
Si pegamos esta URL en un navegador Web, vemos los datos de la tabla “employees”:



The screenshot shows a browser window with the URL <https://qlzlwqmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/>. The page displays a JSON response for two employees. The first employee (ID 100) has a salary of 24000. The second employee (ID 101) has a salary of 100.

```
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://qlzlwqmqrw5rrikd-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”:

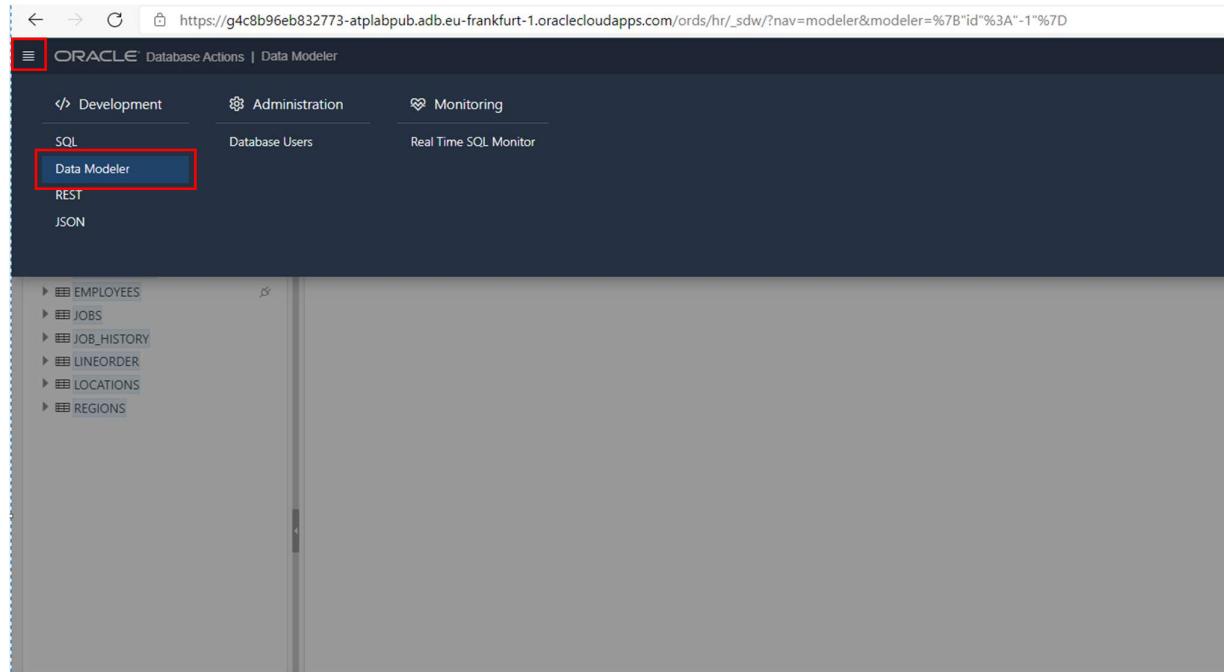


The screenshot shows a browser window with the URL <https://qlzlwqmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/100>. The page displays a JSON response for employee ID 100. The salary is explicitly shown as 24000.

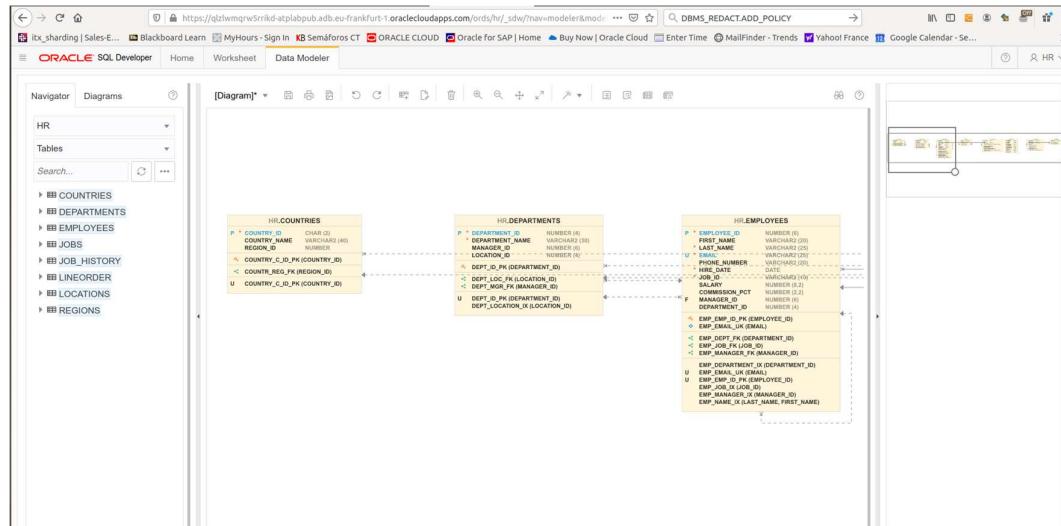
```
employee_id: 100
first_name: "Steven"
last_name: "King"
email: "SKING"
phone_number: "515.123.4567"
hire_date: "2003-06-17T00:00:00Z"
job_id: "AD_PRES"
salary: 24000
commission_pct: null
manager_id: null
department_id: 90
links:
  0:
    rel: "self"
    href: "https://qlzlwqmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/100"
  1:
    rel: "edit"
    href: "https://qlzlwqmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/emp/100"
  2:
    rel: "describedby"
    href: "https://qlzlwqmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/hr/metadata-catalog/emp/item"
  3:
    rel: "collection"
    href: "https://qlzlwqmqrw5rrikd-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.



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**”:

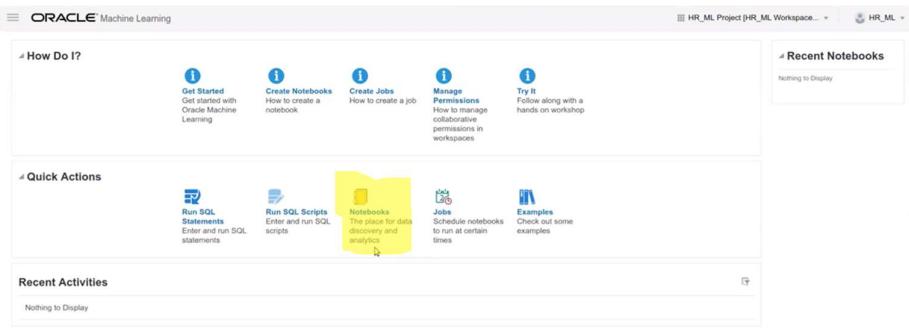
The screenshot shows the Oracle Service Console interface. On the left, there's a sidebar with 'Autonomous Transaction Processing' selected under 'Development'. The main area has several cards: 'Download Oracle Instant Client', 'Oracle APEX', 'Oracle Machine Learning Notebooks' (which is highlighted in red), and 'RESTful Services and SODA'. The 'Oracle Machine Learning Notebooks' card provides a brief description of what it is and how it can be used.

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

The screenshot shows a login form for the Oracle Machine Learning Database. It asks for a 'Database name:' which is set to 'ATPLABPUB'. Below that, 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 input. At the bottom is a blue 'Sign In' button.

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

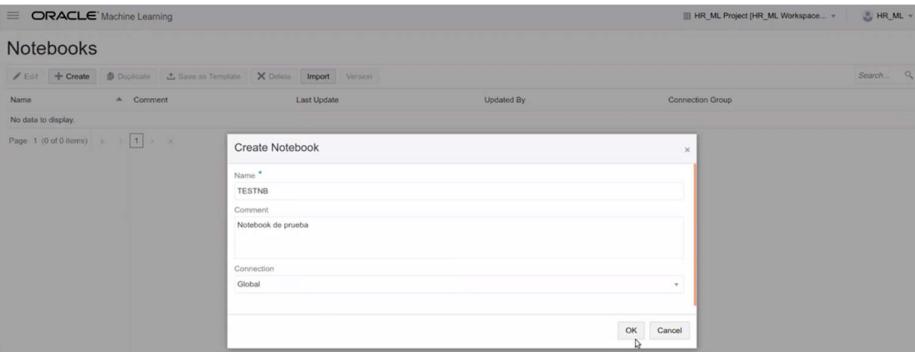




Esto dará paso a la creación de nuestro primer Notebook de Machine Learning. Pulsamos en el botón de crear:



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



A continuación, ejecutamos una query en el nuevo notebook:

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



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



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

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



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

```

select trunc(sum(l_extendedprice)*l_discount) as revenue
from lineitem
where l_orderdate >= date '1994-01-01'
and l_orderdate <= date '1994-01-14'
and l_shipmode between 1 and 2
and l_qty <= 25
and l_extendedprice* l_discount >= 150

```

The screenshot shows the Oracle Machine Learning Notebook interface with a query editor and a results pane. The results pane displays the output of the query, which is a single value: 2611278600347.

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 with a query editor and a results pane. The results pane displays the output of the query, which is a table of employee data. The table has 10 columns: Employee_ID, First_Name, Last_Name, Email, Phone_Number, Hire_Date, Job_Id, Salary, Commission_Pct, and Manager_Id. The salary column is highlighted in red, indicating it is masked.

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

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 with a query editor and a results pane. The results pane displays the output of the query, which is a table of employee data. The salary column is not masked, showing the actual values: 99999, 99999, 99999, 9999, 9999, 9999, 9999, 9999, 9999, and 4200 respectively.

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	17000		100	90
102	Lex	De Haan	LDEHAAN	515.123.4569	2001-01-13T00:00:00Z	AD_VP	17000		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	4200		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”:

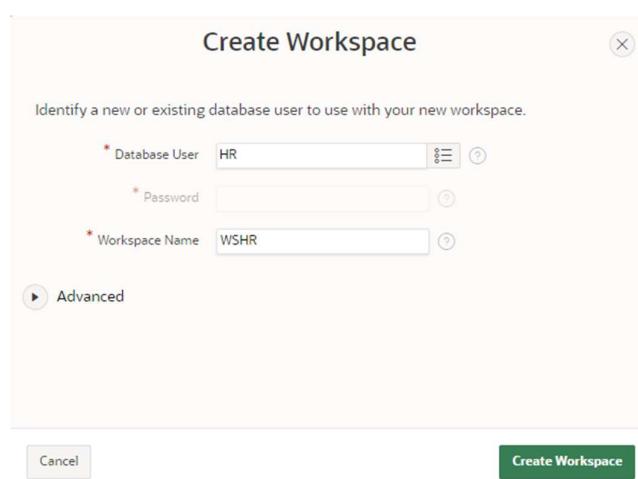
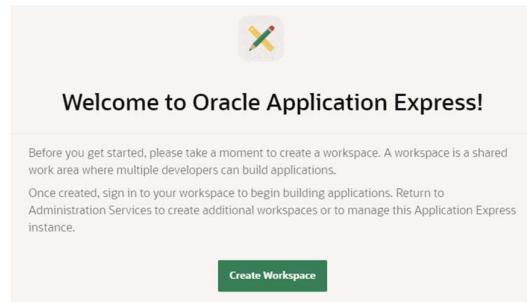
The screenshot shows the Oracle Cloud interface for an Autonomous Database named 'atplabpub'. The 'Tools' tab is highlighted with a red box. The 'Oracle Application Express' section is also highlighted with a red box. The page includes tabs for DB Connection, Performance Hub, Service Console, Scale Up/Down, and More Actions. It displays sections for Database Actions, Oracle ML User Administration, Graph Studio, and SODA Drivers.

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 features a logo of a pencil and a key, and the text 'Administration Services'. Below it, it says 'Welcome to Oracle APEX! Please sign in using the administrator (ADMIN) password of your Autonomous Database.' There is a password input field with a key icon and a 'Sign In to Administration' button. 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 - 한국어.

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



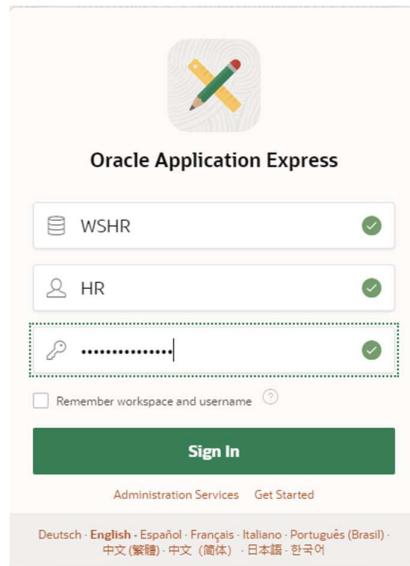


Una vez creado el Workspace, nos conectamos a APEX con el usuario HR, siguiendo el enlace arriba a la izquierda en la pantalla principal de APEX.

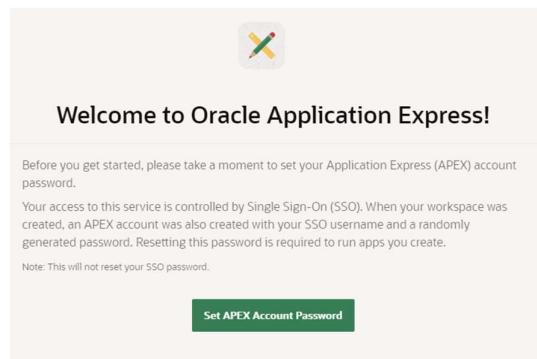
This screenshot shows the Oracle Application Express Administration Services interface in Mozilla Firefox. The top navigation bar includes links for Manage Instance, Manage Workspaces, and Monitor Activity. The main content area displays a "System Message" table with various system logs and a "Jobs" table showing recent database jobs. On the right side, there is a sidebar with sections for About, Instance Tasks, and Workspace Tasks, along with a "Create Workspace" button.

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





Seguimos los pasos siguientes:



Completamos el perfil del usuario HR:

The screenshot shows the "Edit Profile" dialog box. Under "Profile Details", the workspace is set to "WSHR" and the username is "HR". The email address is listed as "opc@mybestdemo.com". There are fields for "First Name" and "Last Name", both currently empty. Under "Profile Photo", there is a placeholder for a photo with a "Choose File" button and a "+" icon. At the bottom of the dialog are "Cancel" and "Apply Changes" buttons.

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 top navigation bar includes links like 'Launch Menu', 'Zimbra', 'Oracle Cloud', 'Autonomous', 'Oracle Database', 'what is my IP', 'DBMS_REDACT_ADD_POLICY', and 'Launch Menu'. The main menu on the left has sections for 'App Builder', 'SQL Workshop' (which is currently selected), 'Team Development', and 'App Gallery'. Under 'SQL Workshop', there are sub-options: 'Object Browser', 'SQL Commands', 'SQL Scripts', 'Utilities', and 'Restful Services'. The central area displays 'Top Apps' (with one entry for 'hr'), 'Top Users' (with one entry for 'hr'), and 'News and Messages'. On the right side, there's a sidebar with sections for 'About', 'Dashboard', 'Resources', and social media links. The URL in the address bar is https://qlzwmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/r?p=4850:100:1056068599745:::RIRP0_SELECTED_NODE:OBO.

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 top navigation bar includes links like 'Launch Menu', 'Zimbra', 'Oracle Cloud', 'Autonomous', 'Oracle Database', 'what is my IP', 'DBMS_REDACT_ADD_POLICY', and 'Launch Menu'. The main menu on the left has sections for 'RESTful Services' (selected) and 'ORDS RESTful Services'. Under 'ORDS RESTful Services', there are sub-options: 'ORDS Version' (19.4.3.r1061746), 'De-Register Schema from ORDS', 'Install Sample Service', 'Import', and 'Configure'. The central area displays 'Schema Access' (Access Status: ENABLED), 'Metadata Access' (Authorization Required: ENABLED), and 'Schema Aliased' (Schema Alias: hr). Below these are sections for 'Modules' (0 Total Modules), 'Privileges' (4 Total Privileges), 'Roles' (8 Total Roles), and 'Enabled Objects' (1 Total Enabled Objects). At the bottom, there are sections for 'Module Status' (No Modules Defined) and 'Object Aliases' (No Modules Defined). The URL in the address bar is https://qlzwmqrw5rrikd-atplabpub.adb.eu-frankfurt-1.oraclecloudapps.com/ords/r?p=4850:190:1056068599745:::RIRP0_SELECTED_NODE:OBO.



The screenshot shows the Oracle APEX interface for managing ORDS enabled objects. The left sidebar has 'ORDS RESTful Services' selected. Under 'Enabled Objects', 'ORDS RESTful Services' is also selected. The main content area displays a table titled 'RESTFUL Enabled Objects'. The table has columns: Parsing Schema, Parsing Object, Object Alias, Type, Status, Auto REST Auth, Ops Allowed, Type Path, and Aliased. One row is shown: HR, EMPLOYEES, emp, TABLE, ENABLED, DISABLED, - (disabled), ENABLED, /HR, and Aliased (with a green checkmark). A legend at the bottom indicates that a green circle means 'Object name and alias are different' and a red circle means 'Object name and alias are the same'.

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

This screenshot shows the Oracle APEX SQL Workshop Object Browser. The left sidebar has 'ORDS RESTful Services' selected. Under 'ORDS RESTful Services', 'Object Browser' is selected. The main content area is identical to the previous screenshot, displaying the 'ORDS Enabled Objects' table with one row: HR, EMPLOYEES, emp, TABLE, ENABLED, DISABLED, - (disabled), ENABLED, /HR, and Aliased (with a green checkmark).



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

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 Yes No [?](#)

Object Alias [?](#)

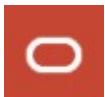
Authorization Required Yes No [?](#)

RESTful URI [Copy](#) [?](#)

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/"}]}
```



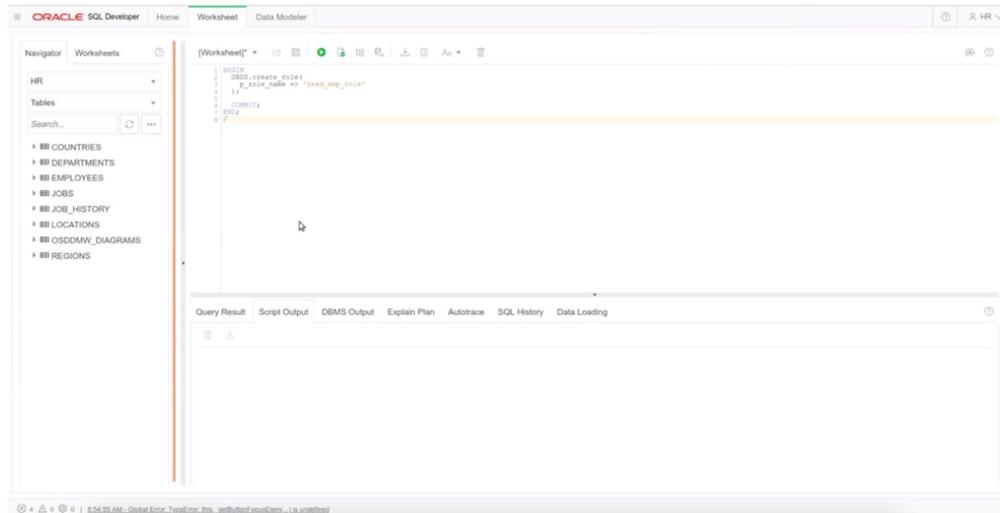
Ejercicio 5 (opcional): Configuración de seguridad de acceso OAuth2

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”:

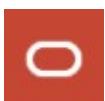
```
BEGIN
    ORDS.create_role(
        p_role_name => 'read_emp_role'
    );
    COMMIT;
END;
/
```



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 with the 'Worksheet' tab selected. In the central workspace, there is a PL/SQL script:

```
1 BEGIN
2   ORDS.create_role(
3     p_role_name => 'read_emp_role'
4   );
5   COMMIT;
6 END;
7 /
8
9 DECLCLASS
10 1.ARR CMA.VC_ARR1;
11 1.ARR(1) := 'read_emp_role';
12
13 ORDS.define_privilege(
14   p_privilege_name => 'read_emp_priv',
15   p_label           => 'Employee reader privilege',
16   p_description     => 'Allow to query employees'
17 );
18
19 1.ARR;
20
21 COMMIT;
22 END;
23 /
24
25 /
26
```

The script creates a role named 'read_emp_role', defines a privilege 'read_emp_priv' with a label 'Employee reader privilege' and description 'Allow to query employees', and then commits the changes.

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

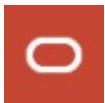
```
SELECT id, name
FROM user_ords_privileges
WHERE name = 'read_emp_priv';

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 with the 'Query Result' tab selected. It displays the results of the two queries run in the previous step:

id	name
1	read_emp_priv

privilege_id	privilege_name	role_id	role_name
1	read_emp_priv	1	read_emp_role



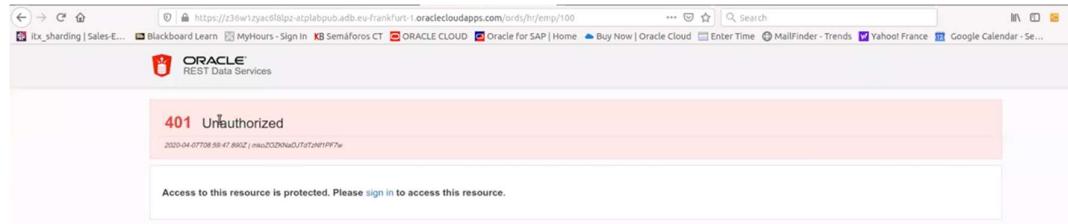
A continuación, mapeamos el privilegio a todas las terminaciones del endpoint “/emp/*”.

```
BEGIN
    ORDS.create_privilege_mapping(
        p_privilege_name => 'read_emp_priv',
        p_pattern => '/emp/*'
    );
    COMMIT;
END;
/
SELECT privilege_id, name, pattern
FROM    user_ords_privilege_mappings
WHERE   name = 'read_emp_priv';
```

The screenshot shows a SQL worksheet interface. At the top, there is a code editor window containing the provided PL/SQL script. Below it is a results window titled 'Query Result' which displays a single row of data from the 'user_ords_privilege_mappings' table. The results table has three columns: 'privilege_id', 'name', and 'pattern'. The data row is: 10061, read_emp_priv, /emp/*.

privilege_id	name	pattern
10061	read_emp_priv	/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 developer interface. In the top worksheet, there is PL/SQL code to create a client and select its details. In the bottom Query Result window, the output of the SELECT statement is shown in a table:

	id	name	client_id	client_secret
1	10064	Employee Client	mVzhiKydBlVdL...	h0R4I0hJE6XV4...

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         => 'Employee reader',
6     p_description   => 'Client app for employees consultation',
7     p_support_email => 'timeexample.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;
16
17 SELECT name, client_name
18 FROM user_ords_client_privileges;
19
20 BEGIN
21   OAUTH.grant_client_role(
22     p_client_name =>'Employee Client',
23     p_role_name   =>'read_emp_role'
24   );
25
26  COMMIT;
27 END;
28 /
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://<use\_su ATP ORDS URL>/ords/hr/oauth/token
```

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

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

