



CICLO: [DAM]
MÓDULO DE [BASES DE DATOS]

[Tarea N° 04]

Alumno:
[Juan Carlos Filter Martín]
[15456141A]

Contenido

- 1. Documentos que se adjuntan a este informe.....3**
- 2. (RA03_e) Se han realizado consultas resumen.....3**
 - Apartado1..... 3
 - Apartado2..... 4
 - Apartado3..... 5
 - Apartado4..... 6
 - Apartado5..... 7
 - Apartado6..... 8
- 3. (RA03_f) Se han realizado consultas con subconsultas.....9**
 - Apartado1..... 9
 - Apartado2..... 10
 - Apartado3..... 11
- 4. (RA03_c) Se han realizado consultas sobre el contenido de varias tablas mediante composiciones internas.....12**
 - Apartado1..... 12
 - Apartado2..... 13
 - Apartado3..... 14
- 5. (RA03_d) Se han realizado consultas sobre el contenido de varias tablas mediante composiciones externas.....15**
 - Apartado1..... 15
 - Apartado2..... 16
 - Apartado3..... 17
 - Apartado4..... 18

1. Documentos que se adjuntan a este informe.

A continuación se detallan los documentos que componen la presente entrega de la tarea:

1. Informe de elaboración de la tarea.
2. Archivo texto plano SQL

2. (RA03_e) Se han realizado consultas resumen.

• → *Apartado1*

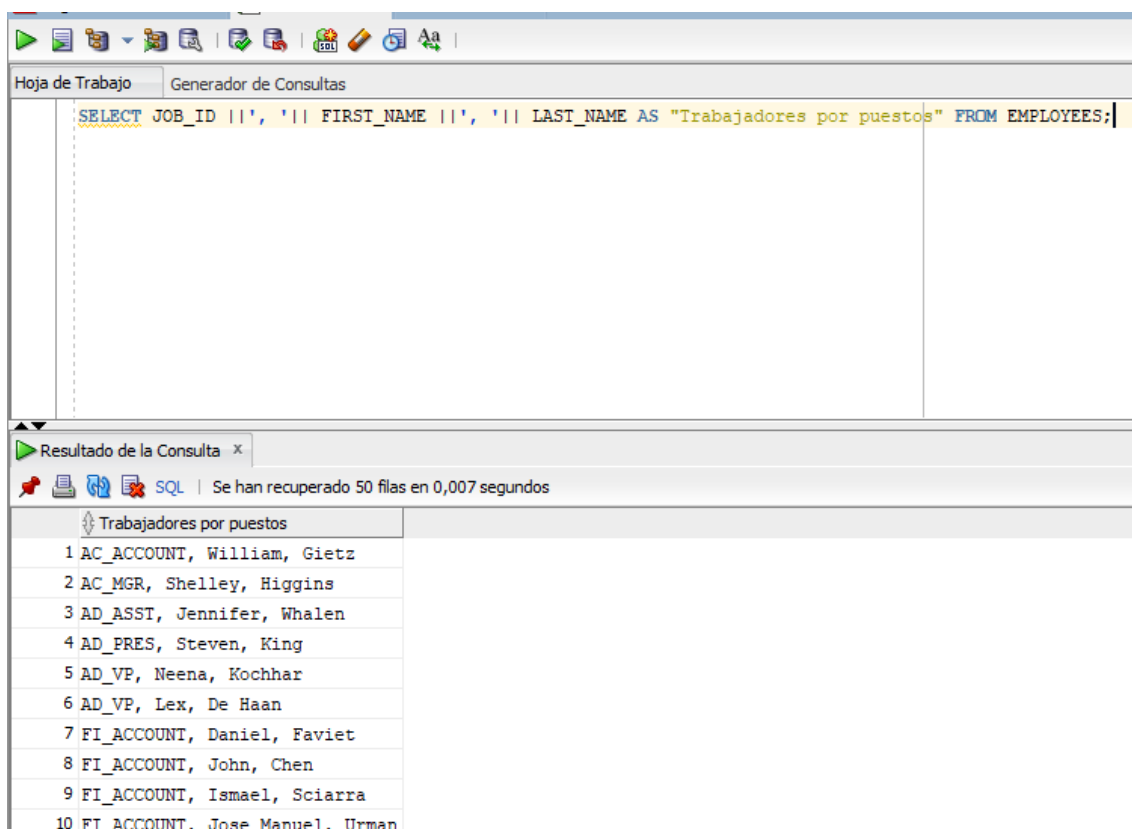
El departamento de recursos humanos ha solicitado un informe de todos los empleados y el id de su cargo (job_id). Muestre con el ID del cargo (job_id) concatenado con el nombre (first_name), apellido (last_name), separados por una coma y un espacio. Asignándole el alias 'Trabajadores por puesto'

Para hacer una consultar es mediante SELECT seguido de los campos que vamos a mostrar y concatenado con ||

Los Alias se escribe entre comillas dobles y es opcional poner AS

Por ultimo le indicamos la tabla mediante un FROM

```
de Trabajo | Generador de Consultas
SELECT JOB_ID || ', ' || FIRST_NAME || ', ' || LAST_NAME AS "Trabajadores por puestos" FROM EMPLOYEES;
```



The screenshot shows a SQL query generator tool with a toolbar at the top. The main window is titled 'Hoja de Trabajo' and 'Generador de Consultas'. It contains the same SQL query as the previous block. Below the query editor, there is a section titled 'Resultado de la Consulta' which shows the results of the query. The results are displayed in a table with 10 rows and 1 column. The first row is the header 'Trabajadores por puestos'. The following rows show the job ID, first name, and last name concatenated with commas and spaces.

Trabajadores por puestos
1 AC_ACCOUNT, William, Gietz
2 AC_MGR, Shelley, Higgins
3 AD_ASST, Jennifer, Whalen
4 AD_PRES, Steven, King
5 AD_VP, Neena, Kochhar
6 AD_VP, Lex, De Haan
7 FI_ACCOUNT, Daniel, Faviot
8 FI_ACCOUNT, John, Chen
9 FI_ACCOUNT, Ismael, Sciarra
10 FI_ACCOUNT, Jose Manuel, Urman

- → *Apartado2*

El departamento de recursos humanos desea una consulta para mostrar todos los nombres (first_name), apellidos (last_name) de los empleados del departamento 80 cuyo nombre empiece por "A".

Le indicamos que muestre el nombre, apellido de la tabla empleados que cumpla la condición que sean del departamento 80 y que empiece por el nombre A mediante Like A %

```
SELECT FIRST_NAME, LAST_NAME FROM EMPLOYEES WHERE DEPARTMENT_ID=80 AND FIRST_NAME LIKE 'A%';
```

The screenshot shows a SQL IDE interface with a query editor and a results pane. The query editor contains the following SQL code:

```
SELECT JOB_ID ||', '|| FIRST_NAME ||', '|| LAST_NAME AS "Trabajadores por puestos" FROM EMPLOYEES;  
SELECT FIRST_NAME, LAST_NAME FROM EMPLOYEES WHERE DEPARTMENT_ID=80 AND FIRST_NAME LIKE 'A%';
```

The results pane shows the output of the second query, displaying a table with two columns: FIRST_NAME and LAST_NAME. The results are as follows:

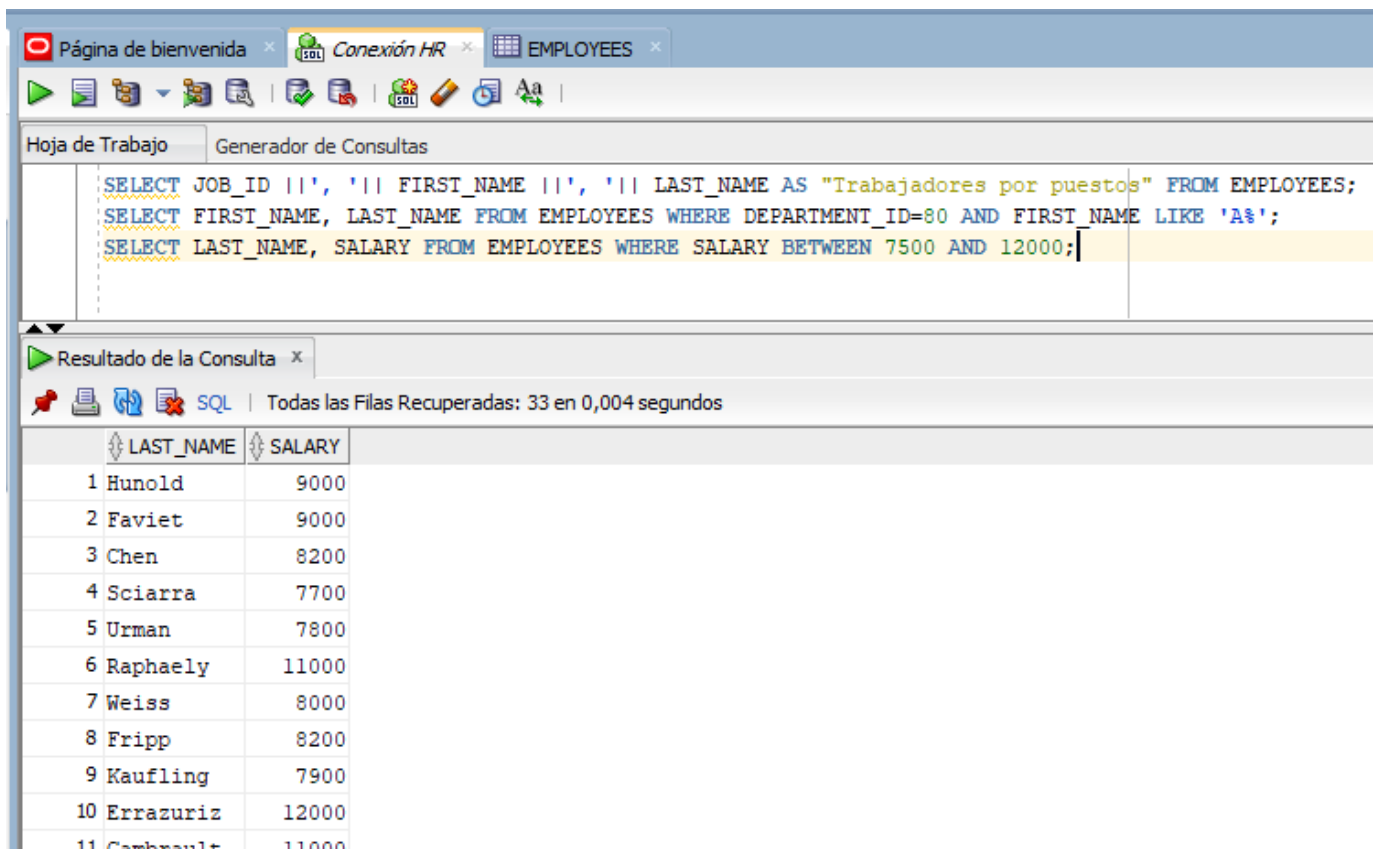
	FIRST_NAME	LAST_NAME
1	Amit	Banda
2	Alberto	Errazuriz
3	Alyssa	Hutton
4	Allan	McEwen

- *Apartado3*

El departamento de recursos humanos necesita encontrar los empleados con salarios ni muy altos ni muy bajos, para ello se debe mostrar el apellido y el salario de cualquier empleado cuyo salario esté entre 7500 y 12000.

Mostramos mediante un SELECT el apellido, el salario de la tabla empleados y que cumpla la condición que el salario sea entre 7500-12000 con Between

```
SELECT LAST_NAME, SALARY FROM EMPLOYEES WHERE SALARY BETWEEN 7500 AND 12000;
```



The screenshot shows the SQL Developer interface. The top pane displays three SQL queries. The third query, highlighted in yellow, is: `SELECT LAST_NAME, SALARY FROM EMPLOYEES WHERE SALARY BETWEEN 7500 AND 12000;`. The bottom pane shows the results of this query in a table with two columns: LAST_NAME and SALARY. The results are sorted by salary in ascending order. The status bar indicates that 33 rows were recovered in 0.004 seconds.

	LAST_NAME	SALARY
1	Hunold	9000
2	Faviet	9000
3	Chen	8200
4	Sciarra	7700
5	Urman	7800
6	Raphaely	11000
7	Weiss	8000
8	Fripp	8200
9	Kaufling	7900
10	Errazuriz	12000
11	Cambrault	11000

- → **Apartado 4**

Encontrar el valor más alto, el valor más bajo, la suma y la media del salario de todos los empleados. Etiquete las columnas como Máximo, Mínimo, Suma y Media, respectivamente.

Para mostrar el máximo, mínimo, suma y media tenemos: MAX(), MIN(), SUM(), AVG().

Tendríamos que poner entre los paréntesis SALARY e indicarle que es de la tabla empleados.

```
SELECT MAX(SALARY) "Máximo", MIN(SALARY) "Mínimo", SUM(SALARY) "Suma", AVG(SALARY) "Media" FROM EMPLOYEES;
```

Hoja de Trabajo		Generador de Consultas		
	<pre>SELECT JOB_ID ', ' FIRST_NAME ', ' LAST_NAME AS "Trabajadores por puestos" FROM EMPLOYEES; SELECT FIRST_NAME, LAST_NAME FROM EMPLOYEES WHERE DEPARTMENT_ID=80 AND FIRST_NAME LIKE 'A%'; SELECT LAST_NAME, SALARY FROM EMPLOYEES WHERE SALARY BETWEEN 7500 AND 12000; SELECT MAX(SALARY) "Máximo", MIN(SALARY) "Mínimo", SUM(SALARY) "Suma", AVG(SALARY) "Media" FROM EMPLOYEES;</pre>			
Resultado de la Consulta x				
SQL Todas las Filas Recuperadas: 1 en 0,003 segundos				
	Máximo	Mínimo	Suma	Media
1	24000	2100	691416	6461,831775700934579439252336448598130841

- → *Apartado5*

Determine el número de jefes (manager_id) que sean distintos. Etiquete la columna como “Numero de Jefes”.

Para ello contamos todo lo que sea distinct manager_id mediante COUNT de la tabla empleado.

Cambiándole a este el nombre mediante un Alias

```
SELECT COUNT(DISTINCT MANAGER_ID) AS "Numero de Jefes" FROM EMPLOYEES;
```

The screenshot shows a SQL IDE interface with the following components:

- Top Bar:** Includes tabs for 'Página de bienvenida', 'Conexión HR', and 'EMPLOYEES'. Below the tabs is a toolbar with icons for running queries, saving, and other database functions.
- Hoja de Trabajo (Worksheet):** Contains the SQL query: `SELECT COUNT(DISTINCT MANAGER_ID) AS "Numero de Jefes" FROM EMPLOYEES;`
- Salida de Script (Script Output):** Shows the status 'Todas las Filas Recuperadas: 1 en 0,002 segundos'.
- Resultado de la Consulta (Query Results):** Displays a table with the following data:

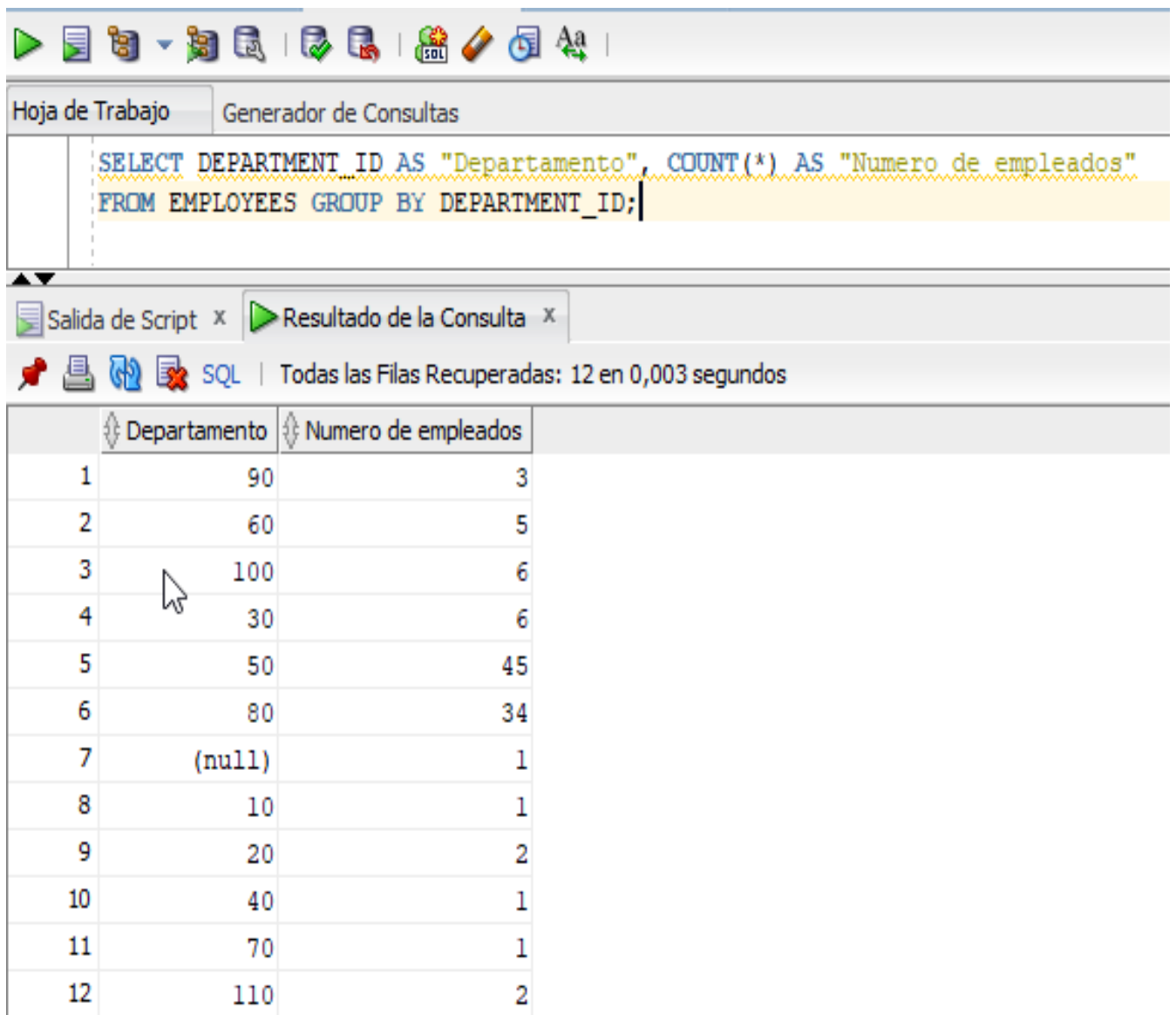
	Numero de Jefes
1	18

- → **Apartado6**

Determina el número de empleados de cada departamento.

Esto va a agrupar todo los department_id que sean iguales y va a mostrar una columna nueva llamada Numero de empleado con el resultado de cuantas filas contiene cada departamento.

```
SELECT DEPARTMENT_ID AS "Departamento", COUNT(*) AS "Numero de empleados"  
FROM EMPLOYEES GROUP BY DEPARTMENT_ID;
```



Hoja de Trabajo | Generador de Consultas

Salida de Script x | Resultado de la Consulta x

Todas las Filas Recuperadas: 12 en 0,003 segundos

	Departamento	Numero de empleados
1	90	3
2	60	5
3	100	6
4	30	6
5	50	45
6	80	34
7	(null)	1
8	10	1
9	20	2
10	40	1
11	70	1
12	110	2

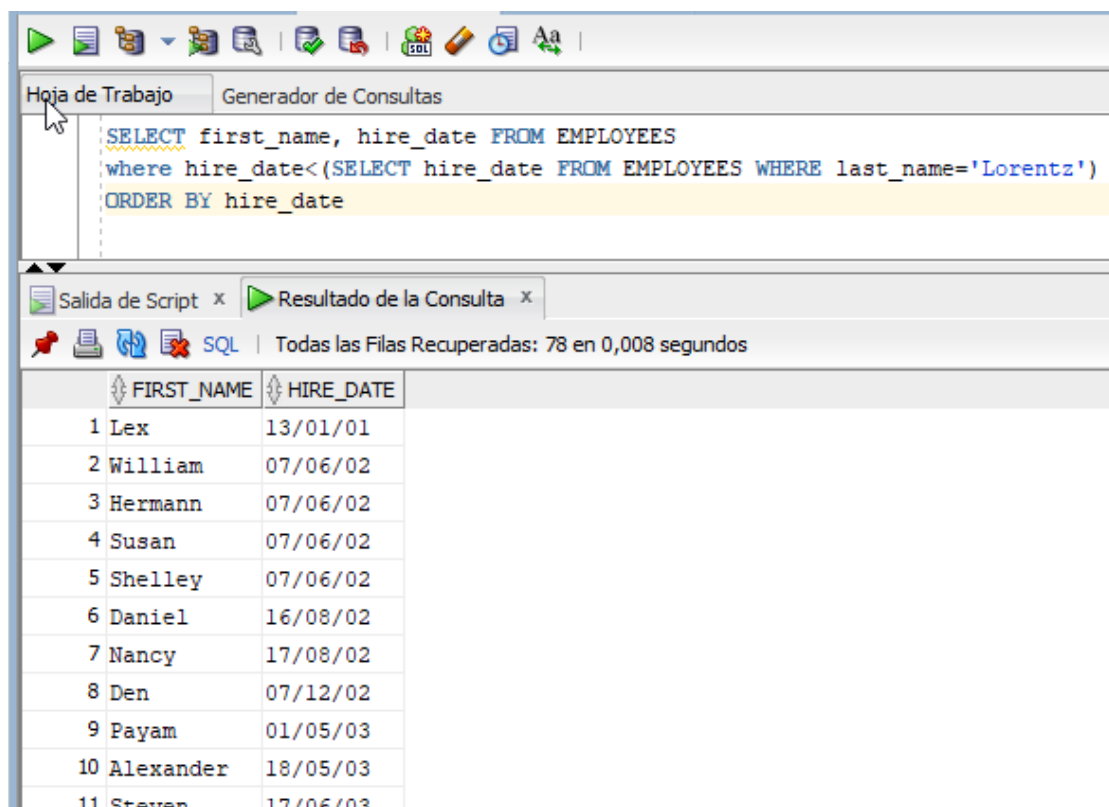
3. (RA03_f) Se han realizado consultas con subconsultas.

- *Apartado1*

El departamento de recursos humanos desea determinar los nombres de todos los empleados contratados antes del empleado con apellido Lorentz. Cree una consulta para mostrar el nombre y la fecha de contratación de cualquier empleado contratado antes de 'Lorentz'.

De esta forma conseguimos mostrar el nombre y la fecha de contratación de la tabla empleados con la condición de que la fecha sea menor que la fecha del apellido 'Lorentz' y en mi caso voy a ordenarlo por fecha mediante un order by aunque no es necesario.

```
SELECT first_name, hire_date FROM EMPLOYEES
where hire_date<(SELECT hire_date FROM EMPLOYEES WHERE last_name='Lorentz')
ORDER BY hire_date
```



The screenshot shows the Oracle SQL Developer interface. The 'Hoja de Trabajo' (Worksheet) tab is active, displaying the SQL query: `SELECT first_name, hire_date FROM EMPLOYEES where hire_date<(SELECT hire_date FROM EMPLOYEES WHERE last_name='Lorentz') ORDER BY hire_date`. The 'Resultado de la Consulta' (Query Result) tab shows the results of the query, which are 78 rows. The first 11 rows are visible in the table below.

	FIRST_NAME	HIRE_DATE
1	Lex	13/01/01
2	William	07/06/02
3	Hermann	07/06/02
4	Susan	07/06/02
5	Shelley	07/06/02
6	Daniel	16/08/02
7	Nancy	17/08/02
8	Den	07/12/02
9	Payam	01/05/03
10	Alexander	18/05/03
11	Steven	17/06/03

Siendo el ultimo con fecha de:

76	Guy	15/11/06
77	Nanette	09/12/06
78	James	14/01/07

Y el empleado con apellido **Lorentz** tiene como fecha:

07 Diana	Lorentz	DLORENTZ	590.423.5567	07/02/07	IT_PR
08 Neena	Kochhar	NEERENDE	515.121.4566	12/03/05	ET_MG

- *Apartado2*

Cree un informe que muestre el id del empleado, nombre, apellido y salario de todos los empleados que ganan más del salario medio. Ordene los resultados en orden ascendente de salario.

Primero mostramos el employee_id, first_name, last_name. Salary de la tabla empleado y que esto cumpla la siguiente condición:

Mediante el where que solamente muestre los salarios mayor al salario medio de la tabla empleados y ordenados ascendente.

```
SELECT EMPLOYEE_ID AS "ID del Empleado", First_name AS "Nombre",  
last_name AS "Apellido", SALARY AS "Salario" FROM employees  
WHERE SALARY > (SELECT AVG(SALARY) FROM employees)  
ORDER BY SALARY ASC;
```

The screenshot shows the SQL Developer interface. At the top, there are tabs for 'Página de bienvenida', 'Conexión HR1.sql', and 'EMPLOYEES'. Below the tabs is a toolbar with various icons. The main window is divided into two panes: 'Hoja de Trabajo' (Worksheet) and 'Generador de Consultas' (Query Editor). The 'Generador de Consultas' pane contains the following SQL query:

```
SELECT EMPLOYEE_ID AS "ID del Empleado", First_name AS "Nombre",  
last_name AS "Apellido", SALARY AS "Salario" FROM employees  
WHERE SALARY > (SELECT AVG(SALARY) FROM employees)  
ORDER BY SALARY ASC;
```

Below the query editor, there are two tabs: 'Salida de Script' (Script Output) and 'Resultado de la Consulta' (Query Result). The 'Resultado de la Consulta' tab is active, showing the results of the query. The results are displayed in a table with the following columns: 'ID del Empleado', 'Nombre', 'Apellido', and 'Salario'. The table contains 5 rows of data:

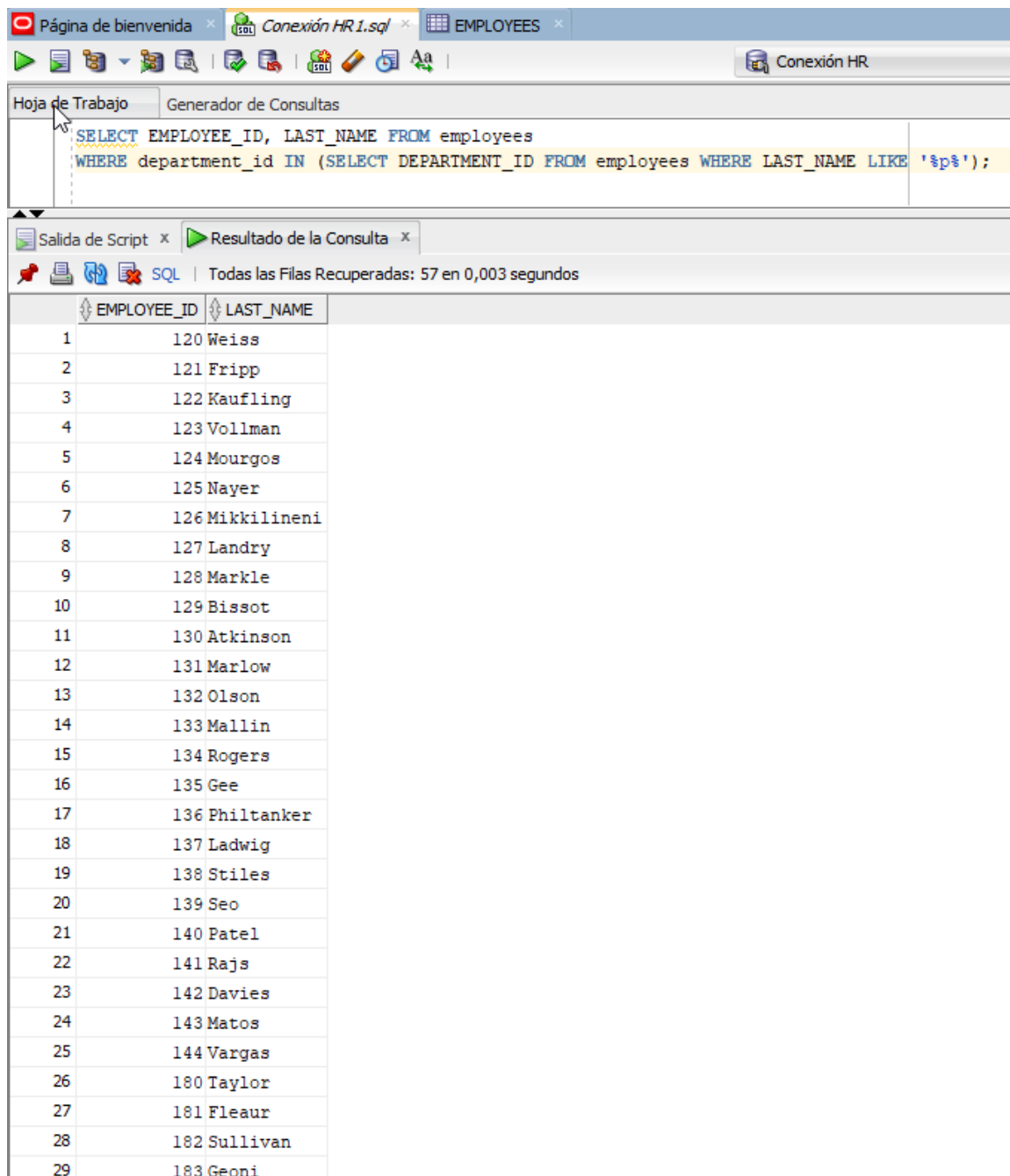
ID del Empleado	Nombre	Apellido	Salario
25	158 Allan	McEwen	9000
26	152 Peter	Hall	9000
27	109 Daniel	Faviet	9000
28	103 Alexander	Hunold	9000
29	157 Patrick	Sully	9500

- *Apartado3*

Escriba una consulta que muestre el id de empleado y el apellido de todos los empleados que trabajan en un departamento en el que hay algún empleado cuyo apellido contiene la letra “p”.

Esta consultar utiliza una subconsulta que obtiene los department_id que contendran en el apellido la “p” y el id de empleados y el apellido que trabajan en esos departamentos.

```
SELECT EMPLOYEE_ID, LAST_NAME FROM employees
WHERE department_id IN (SELECT DEPARTMENT_ID FROM employees WHERE LAST_NAME LIKE '%p%');
```



The screenshot shows the SQL Developer interface. The top pane displays the query: `SELECT EMPLOYEE_ID, LAST_NAME FROM employees WHERE department_id IN (SELECT DEPARTMENT_ID FROM employees WHERE LAST_NAME LIKE '%p%');`. The bottom pane shows the results of the query, which are 29 rows of employee data. The columns are EMPLOYEE_ID and LAST_NAME. The results are as follows:

EMPLOYEE_ID	LAST_NAME
1	120 Weiss
2	121 Fripp
3	122 Kaufling
4	123 Vollman
5	124 Mourgos
6	125 Nayer
7	126 Mikkilineni
8	127 Landry
9	128 Markle
10	129 Bissot
11	130 Atkinson
12	131 Marlow
13	132 Olson
14	133 Mallin
15	134 Rogers
16	135 Gee
17	136 Philtanker
18	137 Ladwig
19	138 Stiles
20	139 Seo
21	140 Patel
22	141 Rajs
23	142 Davies
24	143 Matos
25	144 Vargas
26	180 Taylor
27	181 Fleaur
28	182 Sullivan
29	183 Geoni

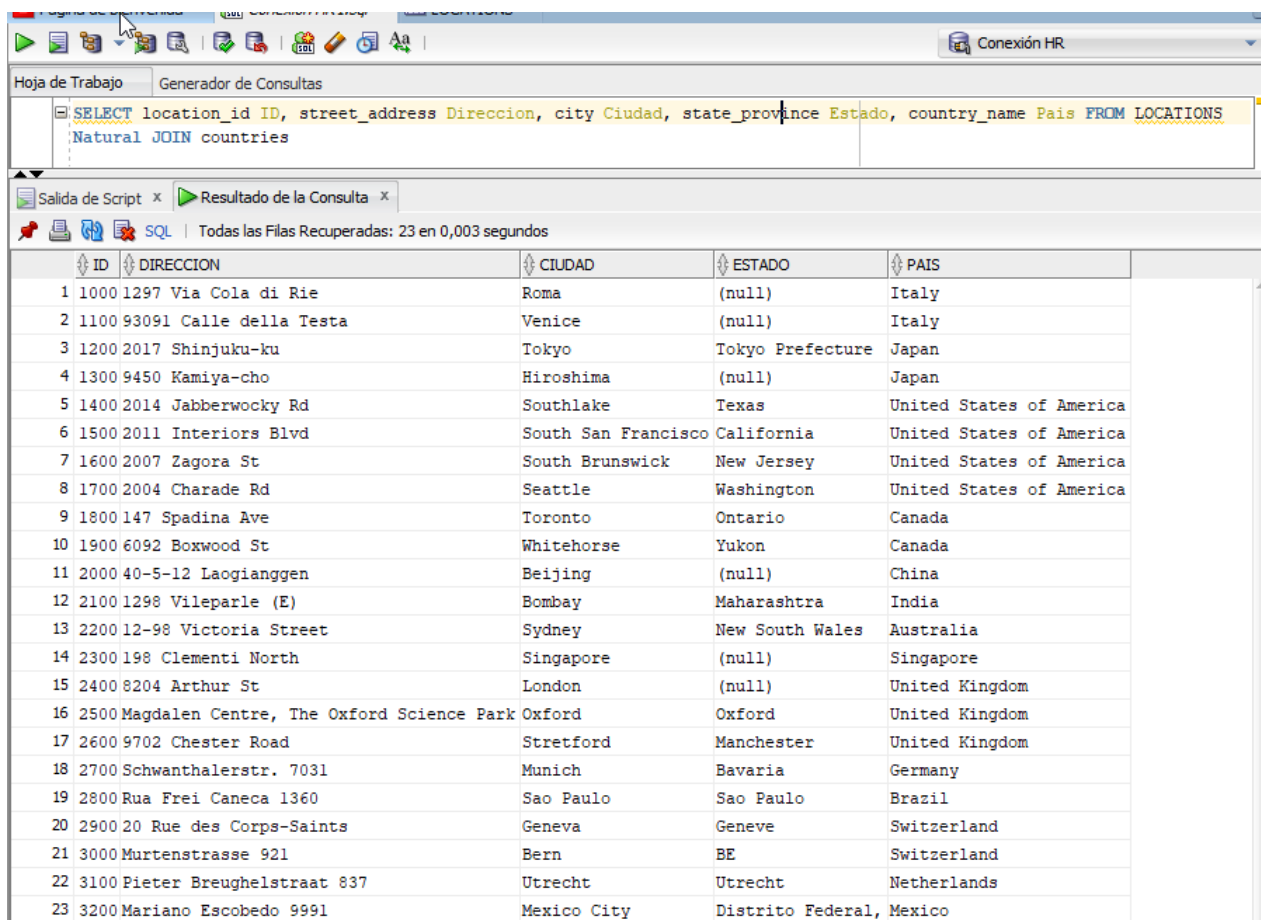
4. (RA03_c) Se han realizado consultas sobre el contenido de varias tablas mediante composiciones internas.

- *Apartado1*

Escriba una consulta para el departamento de recursos humanos que genere las direcciones de todos los departamentos. Utilice las tablas **LOCATIONS y **COUNTRIES**. Muestre el ID, dirección, ciudad, estado y país en la salida. Utilice **NATURAL JOIN** en la consulta.**

Realizamos una consulta para que muestre el id, la dirección, ciudad, estado y país de la tabla **LOCATIONS** seguido del **NATURAL JOIN** **COUNTRIES** para que combine las filas de las tablas con las columnas que tienen el mismo nombre en ambas tablas

```
Hoja de Trabajo | Generador de Consultas
SELECT location_id ID, street_address Direccion, city Ciudad, state_province Estado, country_name Pais FROM LOCATIONS
NATURAL JOIN countries
```



ID	DIRECCION	CIUDAD	ESTADO	PAIS
1	1000 1297 Via Cola di Rie	Roma	(null)	Italy
2	1100 93091 Calle della Testa	Venice	(null)	Italy
3	1200 2017 Shinjuku-ku	Tokyo	Tokyo Prefecture	Japan
4	1300 9450 Kamiya-cho	Hiroshima	(null)	Japan
5	1400 2014 Jabberwocky Rd	Southlake	Texas	United States of America
6	1500 2011 Interiors Blvd	South San Francisco	California	United States of America
7	1600 2007 Zagora St	South Brunswick	New Jersey	United States of America
8	1700 2004 Charade Rd	Seattle	Washington	United States of America
9	1800 147 Spadina Ave	Toronto	Ontario	Canada
10	1900 6092 Boxwood St	Whitehorse	Yukon	Canada
11	2000 40-5-12 Laogianggen	Beijing	(null)	China
12	2100 1298 Vileparle (E)	Bombay	Maharashtra	India
13	2200 12-98 Victoria Street	Sydney	New South Wales	Australia
14	2300 198 Clementi North	Singapore	(null)	Singapore
15	2400 8204 Arthur St	London	(null)	United Kingdom
16	2500 Magdalen Centre, The Oxford Science Park	Oxford	Oxford	United Kingdom
17	2600 9702 Chester Road	Stretford	Manchester	United Kingdom
18	2700 Schwanthalerstr. 7031	Munich	Bavaria	Germany
19	2800 Rua Frei Caneca 1360	Sao Paulo	Sao Paulo	Brazil
20	2900 20 Rue des Corps-Saints	Geneva	Geneve	Switzerland
21	3000 Murtenstrasse 921	Bern	BE	Switzerland
22	3100 Pieter Breughelstraat 837	Utrecht	Utrecht	Netherlands
23	3200 Mariano Escobedo 9991	Mexico City	Distrito Federal, Mexico	

- → *Apartado2*

El departamento de recursos humanos necesita un informe los empleados de varios departamentos. Escriba una consulta para mostrar el nombre, apellido y el nombre de departamento de todos los empleados del departamento 50, 60 y 80.

Realizamos una consulta para que muestre:

- 1 – Nombre, apellido de la tabla empleado con un alias.
- 2 – Nombre de departamento de la tabla con un alias.
- 3 – Con el **JOIN** combinamos las filas de ambas tablas vinculando la consulta empleados con sus departamentos.
- 4 – Por ultimo filtramos la consulta mediante el **WHERE** para que muestre solamente los departamento de los empleados 50, 60 y 80

de Trabajo	Generador de Consultas
	<pre>SELECT E.FIRST_NAME, E.LAST_NAME, D.DEPARTMENT_NAME FROM EMPLOYEES E JOIN DEPARTMENTS D ON E.DEPARTMENT_ID = D.DEPARTMENT_ID WHERE E.DEPARTMENT_ID IN (50,60,80);</pre>

Página de bienvenida × Conexión HR1.sql × LOCATIONS ×

Hoja de Trabajo Generador de Consultas

```
SELECT E.FIRST_NAME, E.LAST_NAME, D.DEPARTMENT_NAME FROM EMPLOYEES E
JOIN DEPARTMENTS D ON E.DEPARTMENT_ID = D.DEPARTMENT_ID
WHERE E.DEPARTMENT_ID IN (50,60,80);
```

Salida de Script × Resultado de la Consulta ×

SQL | Se han recuperado 50 filas en 0,005 segundos

	FIRST_NAME	LAST_NAME	DEPARTMENT_NAME
1	Matthew	Weiss	Shipping
2	Adam	Fripp	Shipping
3	Payam	Kaufling	Shipping
4	Shanta	Vollman	Shipping
5	Kevin	Mourgos	Shipping
6	Julia	Nayer	Shipping
7	Irene	Mikkilineni	Shipping
8	James	Landry	Shipping
9	Steven	Markle	Shipping
10	Laura	Bissot	Shipping
11	Mozhe	Atkinson	Shipping
12	James	Marlow	Shipping
13	TJ	Olson	Shipping
14	Jason	Mallin	Shipping
15	Michael	Rogers	Shipping
16	Ki	Gee	Shipping
17	Hazel	Philtanker	Shipping
18	Renske	Ladwig	Shipping
19	Stephen	Stiles	Shipping

- → *Apartado3*

Muestre el apellido, cargo, número y nombre de departamento de todos los empleados que trabajan en Toronto.

Para ello realizamos la siguiente consulta:

1 - Apellido de la tabla EMPLOYEES

2 - Cargo de la tabla JOBS

3 – Número de la tabla EMPLOYEES

4 – Nombre de departamento de la tabla DEPARTMENTS

5 – Ahora indicamos mediante el **FROM** que la consulta se haga en la tabla EMPLOYEES

6 – Realizamos los **JOIN** de las tablas JOBS y DEPARTMENT para obtener el cargo donde trabaja y el nombre del departamento

7 – Por ultimo filtramos con **WHERE** que contengan en location_id de la tabla departments mediante una subconsulta referenciando a location_id de la tabla LOCATION la ciudad sea 'Toronto'

```
Hoja de Trabajo | Generador de Consultas
SELECT E.LAST_NAME, J.JOB_TITLE, E.EMPLOYEE_ID, D.DEPARTMENT_NAME FROM EMPLOYEES E
JOIN JOBS J ON E.JOB_ID = J.JOB_ID
JOIN DEPARTMENTS D ON E.DEPARTMENT_ID = D.DEPARTMENT_ID
WHERE D.LOCATION_ID IN(SELECT LOCATION_ID FROM LOCATIONS WHERE CITY = 'Toronto');
```

Conexión HR1.sql | JOBS

Hoja de Trabajo | Generador de Consultas

```
SELECT E.LAST_NAME, J.JOB_TITLE, E.EMPLOYEE_ID, D.DEPARTMENT_NAME FROM EMPLOYEES E
JOIN JOBS J ON E.JOB_ID = J.JOB_ID
JOIN DEPARTMENTS D ON E.DEPARTMENT_ID = D.DEPARTMENT_ID
WHERE D.LOCATION_ID IN(SELECT LOCATION_ID FROM LOCATIONS WHERE CITY = 'Toronto');
```

Salida de Script | Resultado de la Consulta

Todas las Filas Recuperadas: 2 en 0,005 segundos

	LAST_NAME	JOB_TITLE	EMPLOYEE_ID	DEPARTMENT_NAME
1	Hartstein	Marketing Manager	201	Marketing
2	Fay	Marketing Representative	202	Marketing

5. (RA03_d) Se han realizado consultas sobre el contenido de varias tablas mediante composiciones externas.

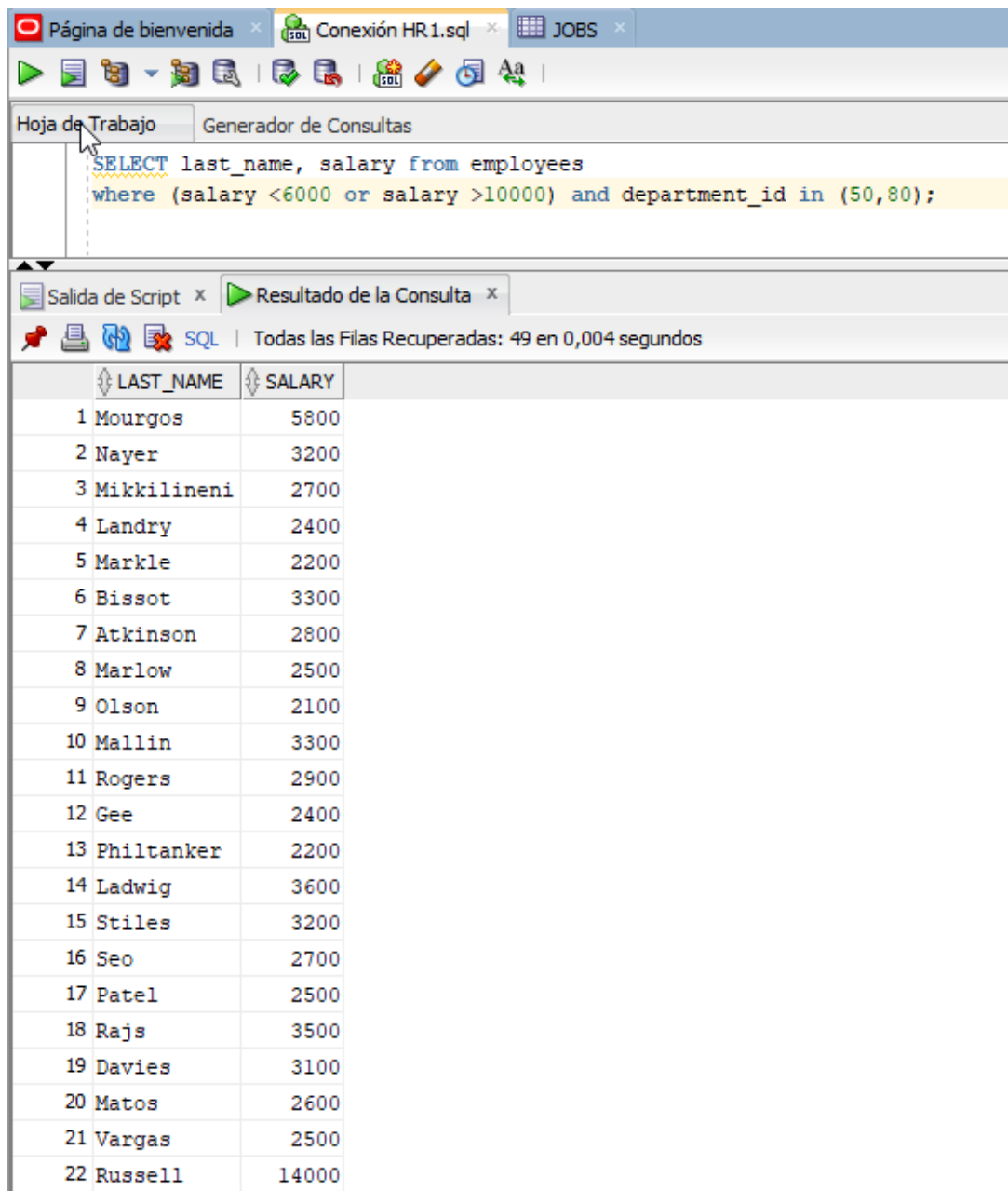
- *Apartado1*

Muestre el apellido y el salario de los empleados que NO ganan entre 6000 y 10000 € y que están en el departamento 50 o 80.

Realizamos una consulta para mostrar el apellido, salario de la tabla empleados y con la condición **WHERE** filtramos que el salario sea (menor que 6000 \leftrightarrow mayor que 10000) y departamento sea 50 y 80.

e Trabajo Generador de Consultas

```
SELECT last_name, salary from employees
where (salary <6000 or salary >10000) and department_id in (50,80);
```



The screenshot shows the SQL Developer interface. At the top, there are tabs for 'Página de bienvenida', 'Conexión HR1.sql', and 'JOBS'. Below the tabs is a toolbar with various icons. The main window is titled 'Hoja de Trabajo' and 'Generador de Consultas'. It contains the same SQL query as shown in the previous block. Below the query editor, there are two tabs: 'Salida de Script' and 'Resultado de la Consulta'. The 'Resultado de la Consulta' tab is active, showing the results of the query. The results are displayed in a table with two columns: 'LAST_NAME' and 'SALARY'. The table contains 22 rows of data, numbered 1 to 22. The salaries range from 2100 to 14000.

	LAST_NAME	SALARY
1	Mourgos	5800
2	Nayer	3200
3	Mikkilineni	2700
4	Landry	2400
5	Markle	2200
6	Bissot	3300
7	Atkinson	2800
8	Marlow	2500
9	Olson	2100
10	Mallin	3300
11	Rogers	2900
12	Gee	2400
13	Philtanker	2200
14	Ladwig	3600
15	Stiles	3200
16	Seo	2700
17	Patel	2500
18	Rajs	3500
19	Davies	3100
20	Matos	2600
21	Vargas	2500
22	Russell	14000

- → *Apartado2*

Muestre los apellidos de los empleados que tengan una “m” en cualquier parte antes de una “a” en su apellido.

Realizamos una consulta para mostrar el apellido de la tabla empleados con where filtrando que el apellido contenga %m%a%

e Trabajo Generador de Consultas

```
SELECT last_name as Apellido from employees where last_name Like 'm%a%';
```

Conexión HR1.sql JOBS

Hoja de Trabajo Generador de Consultas

```
SELECT last_name as Apellido from employees where last_name Like 'm%a%';
```

Salida de Script Resultado de la Consulta

Todas las Filas Recuperadas: 6 en 0,002 segundos

	APELLIDO
1	Cambrault
2	Cambrault
3	Colmenares
4	Kumar
5	Urman
6	Vollman

- *Apartado3*

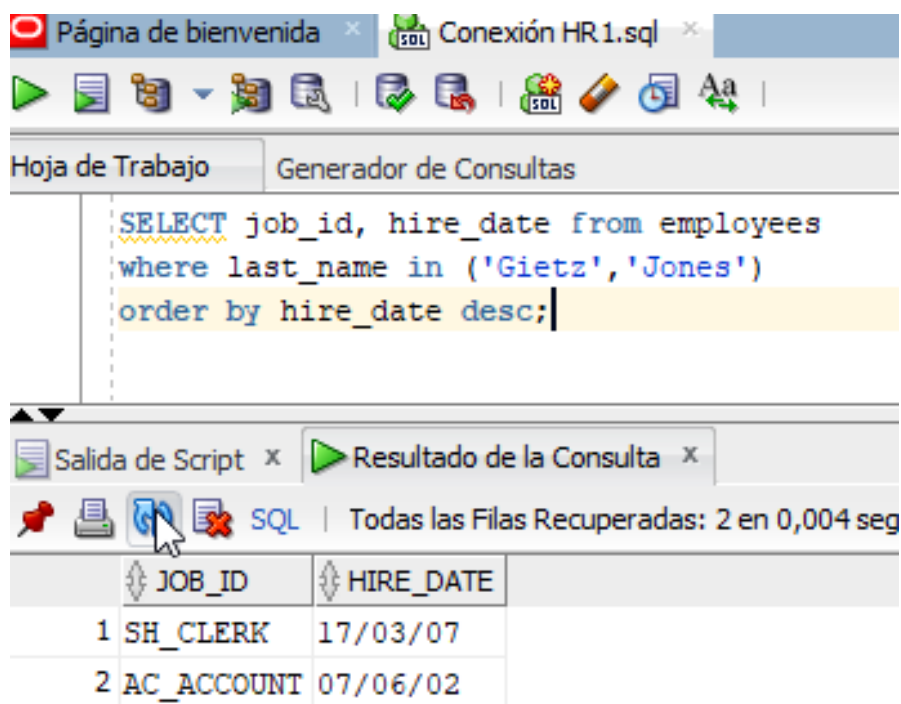
Cree un informe para mostrar el apellido, ID del cargo y fecha de contratación de los empleados cuyos apellidos sean Gietz y Jones. Ordene la consulta en orden descendente por fecha de contratación.

Realizamos una consulta para mostrar el apellido, el job_id y el hire_date de la tabla empleados de la tabla empleados con la filtrando mediante **WHERE** que el apellido sea 'Gietz' y 'Jones' y lordenamos con el **ORDER BY** que la fecha sea descendente.

```

Hoja de Trabajo | Generador de Consultas
SELECT job_id, hire_date from employees
where last_name in ('Gietz','Jones')
order by hire_date desc;

```



Podemos comprobar que con la fecha 17/03/07 es Jones

y con la fecha 07/06/02 es Gietz

95	194 Samuel	MCCAIN	SMCCAIN	650.501.3076	01/07/06	SH_CLERK
96	195 Vance	Jones	VJONES	650.501.4876	17/03/07	SH_CLERK
97	196 Wilson	WATSON	AWATSON	650.507.6911	21/03/06	SH_CLERK
3	205 Shelley	Higgins	SHIGGINS	515.123.8080	07/06/02	
4	206 William	Gietz	WGIETZ	515.123.8181	07/06/02	

- → *Apartado4*

Escriba una consulta para mostrar el nombre, apellido y el nombre de departamento de todos los empleados incluyendo los que no tienen un departamento asignado, pero sin mostrar los departamentos que no tengan empleados.

Para ello realizamos la siguiente consulta:

- 1 – En primer lugar mostramos First_name, Last_name de la tabla empleado.
- 2 – Utilizando un **LEFT JOIN** DEPARTMENTS para recuperar todas las filas que cumplen la condición de igualdad entre la tabla departments y employees y además que muestre todas las filas que no cumplen la condición en la fila izquierda (E.DEPARTMENT_ID)
- 4 – Filtrando con **WHERE** los resultados para excluir los departamentos que no tienen empleados mediante **E.DEPARTMENT_ID IS NULL** incluyendo a los empleados sin departamento asignados y con **OR EMP.DEPARTMENT_ID IN (SELECT DISTINCT DEPARTMENT_ID FROM EMPLOYEES)** Asegurándose esta que solo se incluyan los departamentos que tienen al menos un empleado.

```
SELECT E.FIRST_NAME AS Nombre, E.LAST_NAME AS Apellido
FROM EMPLOYEES E
LEFT JOIN DEPARTMENTS D ON E.DEPARTMENT_ID = D.DEPARTMENT_ID
WHERE E.DEPARTMENT_ID IS NULL OR E.DEPARTMENT_ID IN(SELECT DISTINCT DEPARTMENT_ID FROM EMPLOYEES);
```

The screenshot shows the SQL Developer interface. The top pane displays the query: `SELECT E.FIRST_NAME AS Nombre, E.LAST_NAME AS Apellido FROM EMPLOYEES E LEFT JOIN DEPARTMENTS D ON E.DEPARTMENT_ID = D.DEPARTMENT_ID WHERE E.DEPARTMENT_ID IS NULL OR E.DEPARTMENT_ID IN(SELECT DISTINCT DEPARTMENT_ID FROM EMPLOYEES);`. The bottom pane shows the results of the query in a table with two columns: NOMBRE and APELLIDO. The results list 20 employees, each with their first and last names.

	NOMBRE	APELLIDO
79	Kimberely	Grant
80	Charles	Johnson
81	Winston	Taylor
82	Jean	Fleaur
83	Martha	Sullivan
84	Girard	Geoni
85	Nandita	Sarchand
86	Alexis	Bull
87	Julia	Dellinger
88	Anthony	Cabrio
89	Kelly	Chung
90	Jennifer	Dilly
91	Timothy	Gates
92	Randall	Perkins
93	Sarah	Bell
94	Britney	Everett
95	Samuel	McCain
96	Vance	Jones
97	Alana	Walsh
98	Kevin	Feeney