

ciclo: [DAM]

MÓDULO DE [BASES DE DATOS]

[Tarea Nº 04]

Alumno:

[Juan Carlos Filter Martín]

[15456141A]

**Contenido**

[1. Documentos que se adjuntan a este informe. 3](#__RefHeading___Toc121_13357877)

[2. (RA03\_e) Se han realizado consultas resumen. 3](#__RefHeading___Toc123_13357877)

[→ *Apartado1* 3](#__RefHeading___Toc125_13357877)

[→ *Apartado2* 4](#__RefHeading___Toc125_13357877_Copia_1)

[→ Apartado3 5](#__RefHeading___Toc129_13357877)

[→ Apartado4 6](#__RefHeading___Toc131_13357877)

[→ Apartado5 7](#__RefHeading___Toc131_13357877_Copia_1)

[*→ Apartado6* 8](#__RefHeading___Toc131_13357877_Copia_1_C)

[3. (RA03\_f) Se han realizado consultas con subconsultas. 9](#__RefHeading___Toc133_13357877)

[→ *Apartado1* 9](#__RefHeading___Toc125_13357877_Copia_1_2)

[→ *Apartado2* 10](#__RefHeading___Toc125_13357877_Copia_1_C)

[→ *Apartado3* 11](#__RefHeading___Toc125_13357877_Copia_1_1)

[4. (RA03\_c) Se han realizado consultas sobre el contenido de varias tablas mediante composiciones internas. 12](#__RefHeading___Toc135_13357877)

[→ *Apartado1* 12](#__RefHeading___Toc125_13357877_Copia_1_3)

[→ *Apartado2* 13](#__RefHeading___Toc125_13357877_Copia_1_4)

[→ *Apartado3* 14](#__RefHeading___Toc125_13357877_Copia_1_5)

[5. (RA03\_d) Se han realizado consultas sobre el contenido de varias tablas mediante composiciones externas. 15](#__RefHeading___Toc137_13357877)

[→ *Apartado1* 15](#__RefHeading___Toc125_13357877_Copia_1_6)

[→ *Apartado2* 16](#__RefHeading___Toc125_13357877_Copia_1_7)

[→ *Apartado3* 17](#__RefHeading___Toc125_13357877_Copia_1_7)

[→ *Apartado4* 18](#__RefHeading___Toc125_13357877_Copia_1_8)

# 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

# (RA03\_e) Se han realizado consultas resumen.

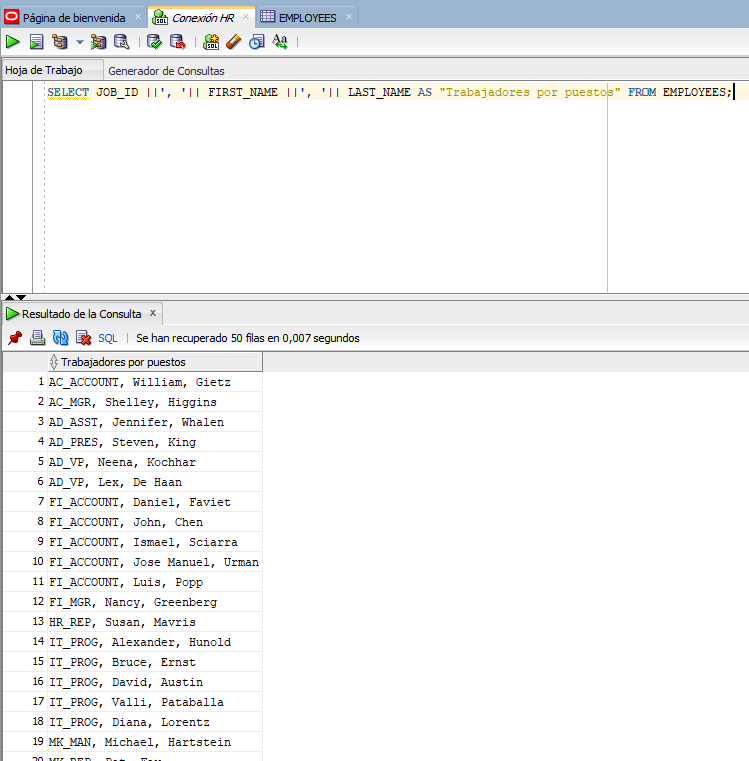
## **→ *Apartado1***

**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

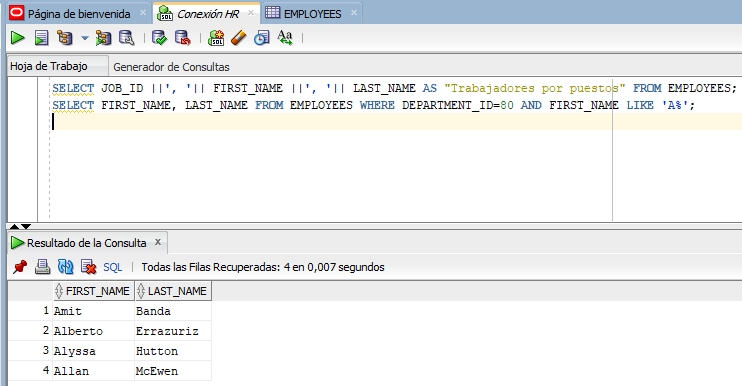
Por ultimo le indicamos la tabla mediante un FROM



## **→ *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%

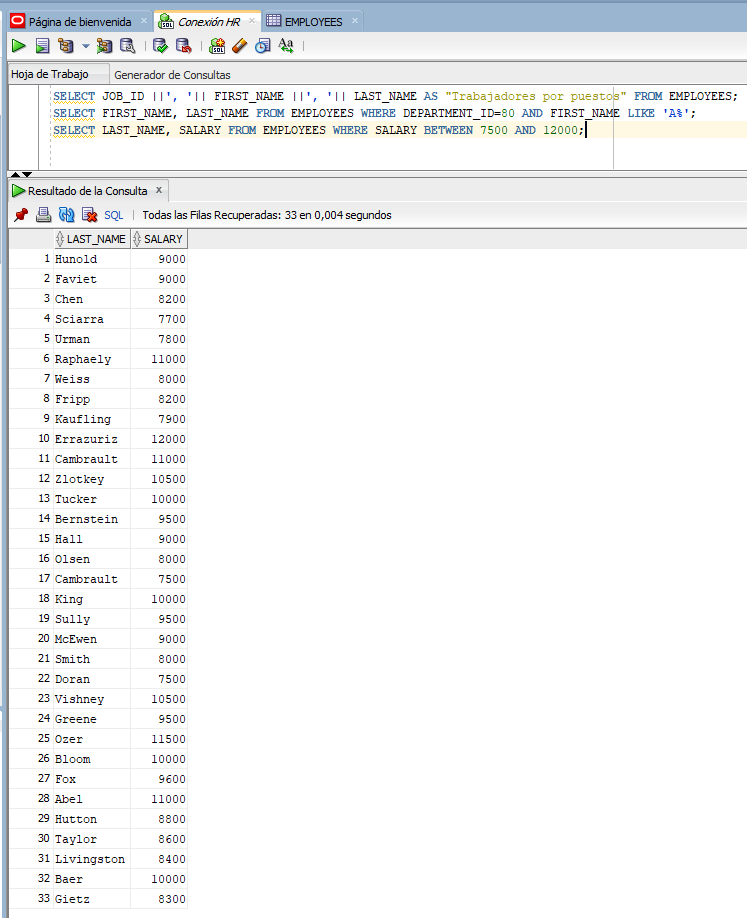


## → Apartado3

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

**



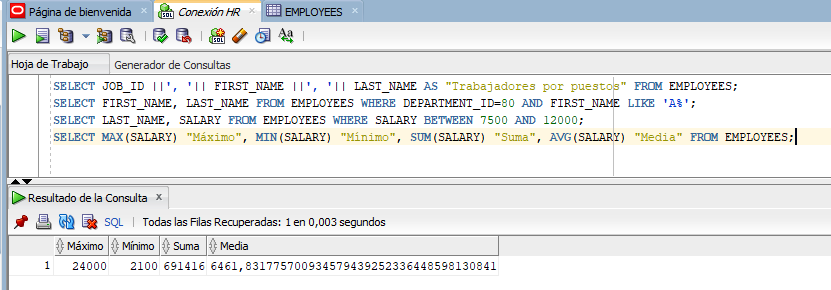
## → Apartado4

**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 maximo, minimo, suma y media tenemos: MAX(), MIN(), SUM(), AVG().

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





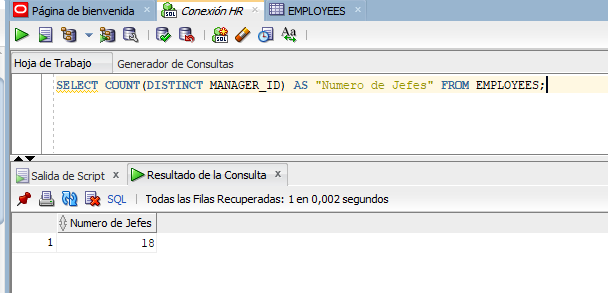
## → Apartado5

**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

**

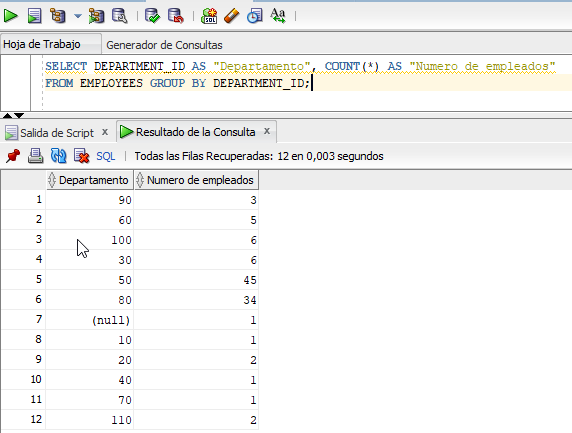
**

## ***→ 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.

**

**

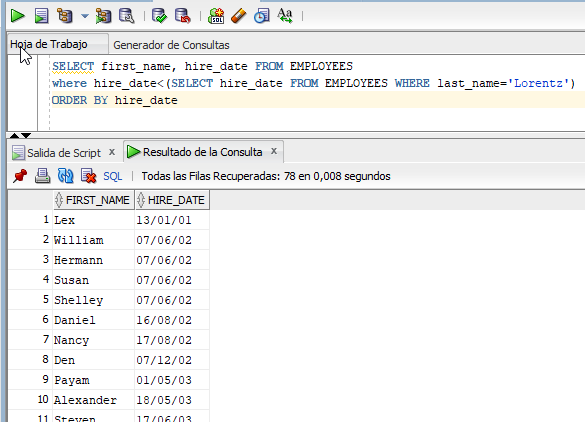
# (RA03\_f) Se han realizado consultas con subconsultas.

## **→ *Apartado1***

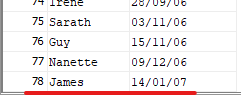
**El departamento de recursos humanos desea determinar los nombres de todos los empleados contratados antes del empleado con apellido Lorentz. Cree una consulta para mostrar el nombre y la fecha de contratación de cualquier empleado contratado antes de 'Lorentz'.**

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.





Siendo el ultimo con fecha de:



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

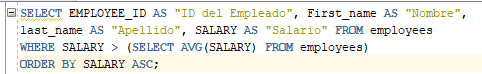


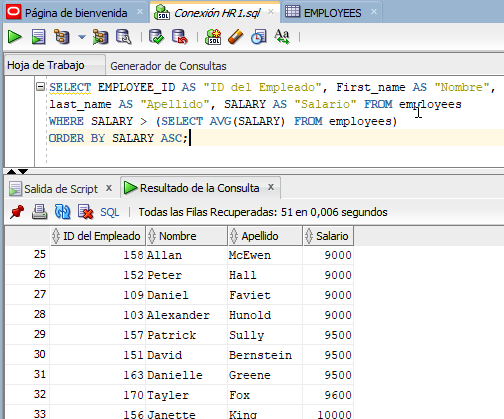
## **→ *Apartado2***

**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.



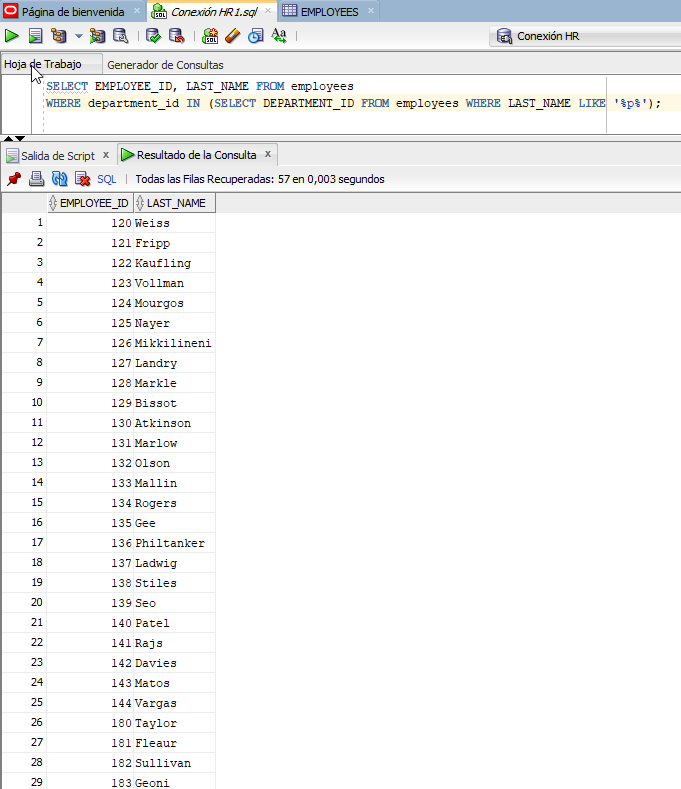


## **→ *Apartado3***

**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.



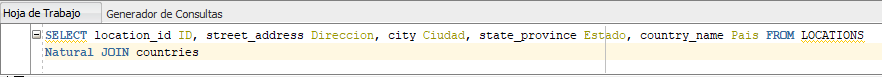


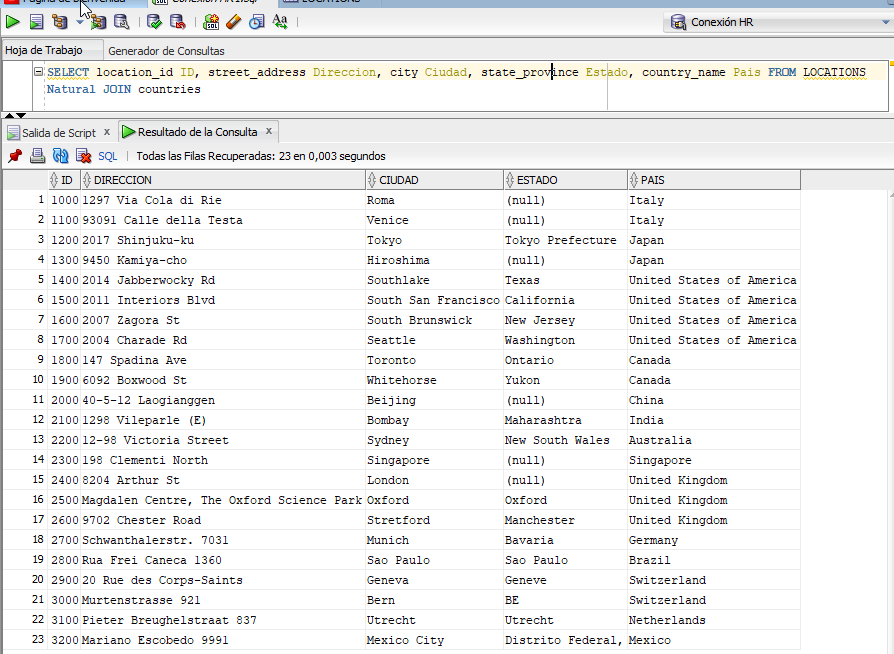
# (RA03\_c) Se han realizado consultas sobre el contenido de varias tablas mediante composiciones internas.

## **→ *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





## **→ *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.**

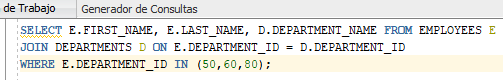
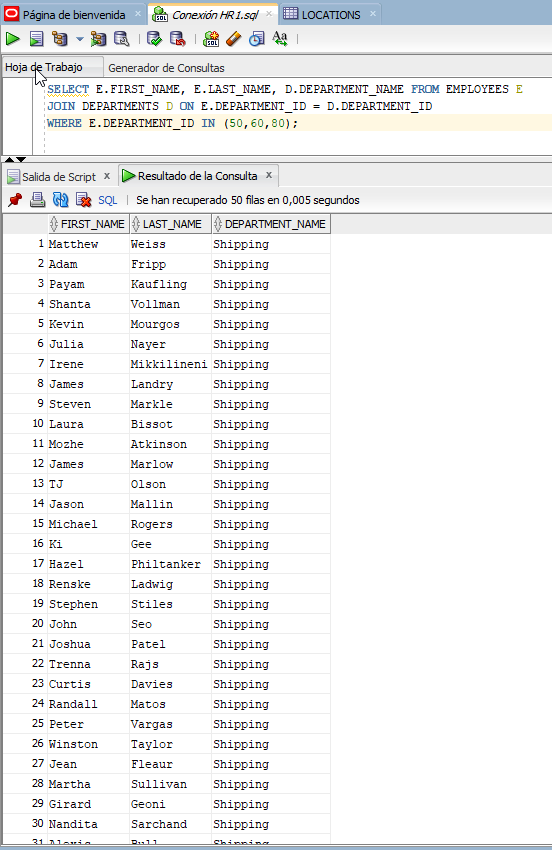
Realizamos una consulta para que muestre:

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



## **→ *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

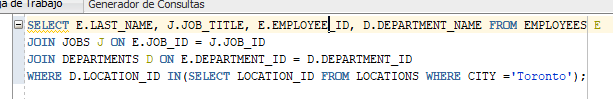
3 – Número de la tabla EMPLOYEES

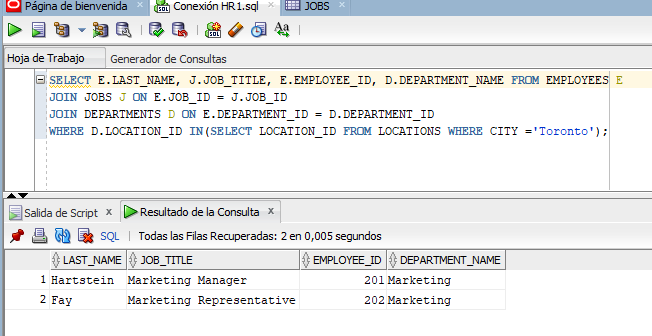
4 – Nombre de departamento de la tabla DEPARTMENTS

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’



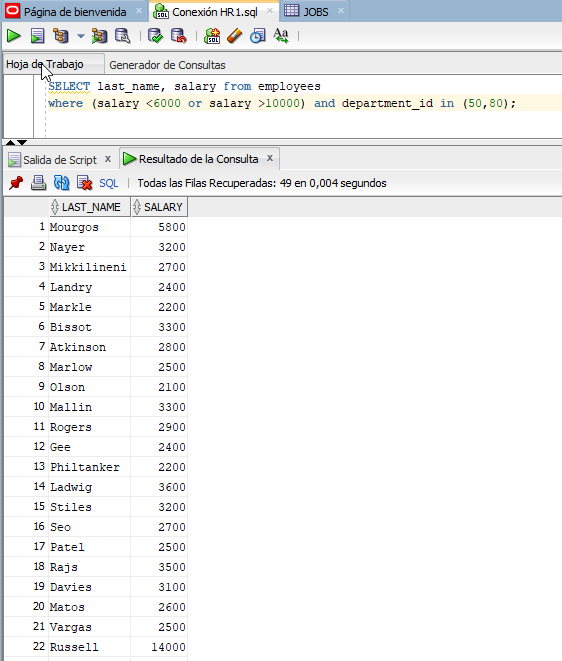


# (RA03\_d) Se han realizado consultas sobre el contenido de varias tablas mediante composiciones externas.

## **→ *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 ←→ mayor que 10000) y departamento sea 50 y 80.

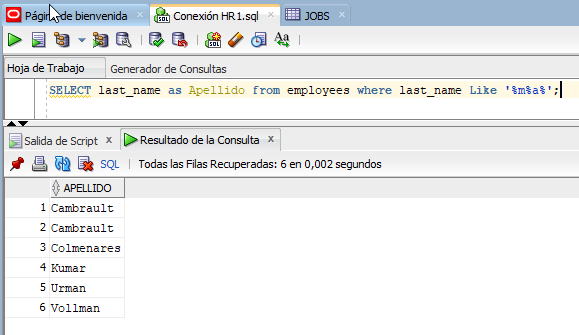


## **→ *Apartado2***

**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%

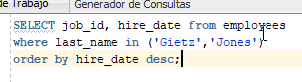


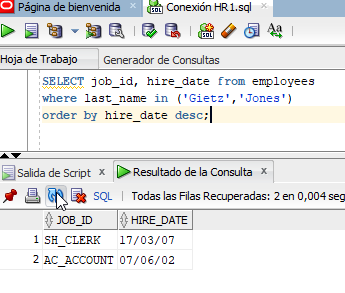


## **→ *Apartado3***

**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.





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

y con la fecha 07/06/02 es Gietz





## **→ *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 ademas 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.

