



Práctica Bases de Datos Avanzadas
Bases de datos paralelas

Breidy Núñez | 2017-5633

Felipe Ramírez | 2018-6104

Luis Franco | 2018-6342

Prof. Leandro Fondeur

20/6/2019

Luego de realizar un estudio del capítulo 20 del libro de texto que cubre el Tema 5 - Bases de Datos Paralelas, responda las siguientes preguntas:

1. ¿Qué es el intercambio de operadores en el paralelismo entre operaciones?

*El **intercambio de operadores**, es el modelo que usa implementaciones existentes de operaciones, actuando sobre copias locales de los datos, junto con una operación de intercambio que traslada los datos entre diferentes procesadores.*

2. Dentro del paralelismo entre consultas, ¿qué es la coherencia caché?

***Coherencia cache**, es cuando un procesador tiene acceso a los datos o los actualiza, el sistema de bases de datos debe asegurar que el procesador tenga la última versión de éstos en su memoria intermedia.*

3. Al hablar de la clasificación del sesgo, ¿a qué se refiere el sesgo de la división?

*Se refiere al hecho de que puede haber un **desequilibrio en la carga de la división**, aunque no haya sesgo en los atributos.*

4. Al hablar de la clasificación del sesgo, ¿a qué se refiere el sesgo de los valores de los atributos?

*El **sesgo de los valores de los atributos** se refiere a que es posible que algunos valores aparezcan en atributos de división pertenecientes a varias tuplas.*

5. Dentro de las técnicas de división, ¿qué es la división por rangos?

*La **división por rangos** consiste en dividir los valores mediante rangos contiguos, lo que es decir sucesiones consecutivas de los valores en distintos discos.*

6. Dentro de las técnicas de división, ¿qué es la división por asociación?

*La **división por asociación** consiste en acoger atributos específicos como factores funcionales en la división de los valores, que se harán en relación a los atributos seleccionados.*

7. Dentro de las técnicas de división, ¿qué es la división por turno rotatorio?

*El esquema de **turno rotatorio** asegura una distribución homogénea de las tuplas entre los discos por ello, cada disco tiene aproximadamente el mismo número de tuplas que los demás. ACCESO Con este esquema tanto las consultas concretas como las de rango son difíciles de procesar, dado que se debe emplear en la búsqueda cada uno de los n discos.*

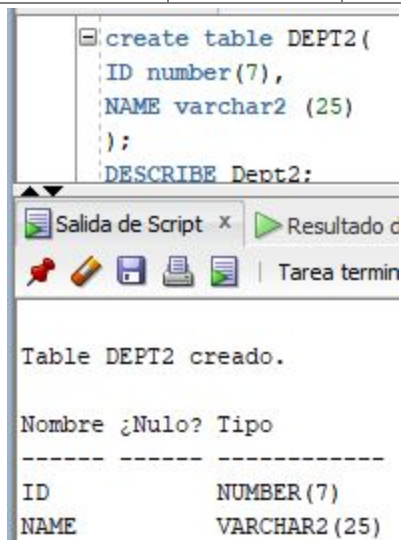
8. ¿Qué es el paralelismo en las Bases de Datos?

*De forma general el concepto de **paralelismo** en las bases de datos lo podríamos definir como la partición de la base de datos (normalmente a nivel de relaciones) para poder procesar de forma paralela en distintos discos y con distintos procesadores una sola operación sobre la base de datos.*

Realice los siguientes ejercicios luego de estudiar el documento Les02-Managing Schema.ppt:

1. Cree la tabla DEPT2 basada en el siguiente gráfico de instancia de tabla. Confirme que la tabla se ha creado.

Column Name	ID	NAME
Data type	NUMBER	VARCHAR2
Length	7	25



```
create table DEPT2(  
  ID number(7),  
  NAME varchar2 (25)  
);  
DESCRIBE Dept2;
```

Salida de Script x Resultado c Tarea termin

Table DEPT2 creado.

Nombre	¿Nulo?	Tipo
ID		NUMBER(7)
NAME		VARCHAR2(25)

2. Rellene la tabla DEPT2 con datos de la tabla DEPARTMENTS. Incluya sólo las columnas que necesite.



3. Cree la tabla EMP2 basada en el siguiente gráfico de instancia de tabla. Confirme que la tabla se ha creado.

Column Name	ID	LAST_NAME	FIRST_NAME	DEPT_ID
Data type	NUMBER	VARCHAR2	VARCHAR2	NUMBER
Length	7	25	25	7

```
create table Emp2 (  
  ID number(7),  
  LAST_NAME varchar2 (25),  
  FIRST_NAME varchar2 (25),  
  DEPT_ID number(7)  
);  
DESCRIBE emp2;
```

Salida de Script x Resultado de la Co

Tarea terminada en

Table EMP2 creado.

Nombre	¿Nulo?	Tipo
ID		NUMBER(7)
LAST_NAME		VARCHAR2(25)
FIRST_NAME		VARCHAR2(25)
DEPT_ID		NUMBER(7)

4. Modifique la tabla EMP2 para permitir apellidos de empleado más largos (50 caracteres). Confirme la modificación.

```
ALTER TABLE emp2
MODIFY (last_name VARCHAR2(50));
DESCRIBE emp2;
```

Salida de Script x Resultado de la Consulta x

Tarea terminada en 0.099 seg

Table EMP2 alterado.

Nombre	¿Nulo?	Tipo
ID		NUMBER(7)
LAST_NAME		VARCHAR2(50)
FIRST_NAME		VARCHAR2(25)
DEPT_ID		NUMBER(7)

5. Confirme que las tablas DEPT2 y EMP2 se han almacenado en el diccionario de datos.

```
SELECT * FROM all_catalog where table_name = 'DEPT2' OR table_name = 'EMP2';
```

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 2 en 0.023 segundos

	OWNER	TABLE_NAME	TABLE_TYPE
1	HR	EMP2	TABLE
2	HR	DEPT2	TABLE

6. Cree la tabla EMPLOYEES2 basándose en la estructura de la tabla EMPLOYEES. Incluya sólo las columnas EMPLOYEE_ID, FIRST_NAME, LAST_NAME, SALARY y DEPARTMENT_ID. Asigne a las columnas de la nueva tabla los nombres ID, FIRST_NAME, LAST_NAME, SALARY y DEPT_ID, respectivamente.

```
create table EMPLOYEES2
as SELECT employee_id as "ID", first_name as "FIRST_NAME",
last_name as "LAST_NAME", salary as "SALARY",
department_id as "DEPT_ID" FROM employees;
DESCRIBE employees2;
```

Salida de Script x Resultado de la Consulta x

Tarea terminada en 0.405 segundos

Table EMPLOYEES2 creado.

Nombre	¿Nulo?	Tipo
ID		NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
SALARY		NUMBER(8,2)
DEPT_ID		NUMBER(4)

7. Borre la tabla EMP2.

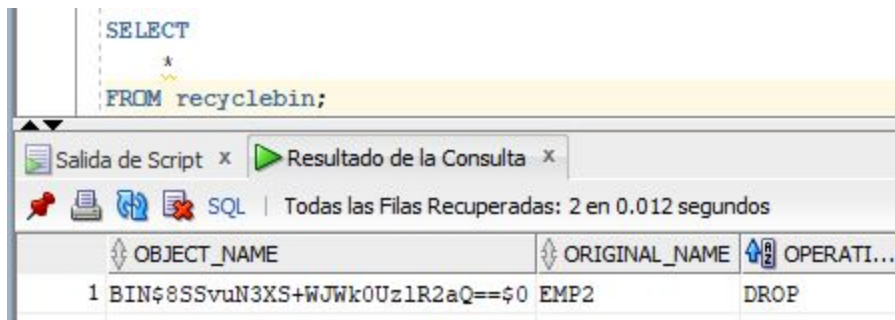
```
DROP TABLE emp2;
```

Salida de Script x Resultado de la Consulta x

Tarea terminada en 0.405 segundos

Table EMP2 borrado.

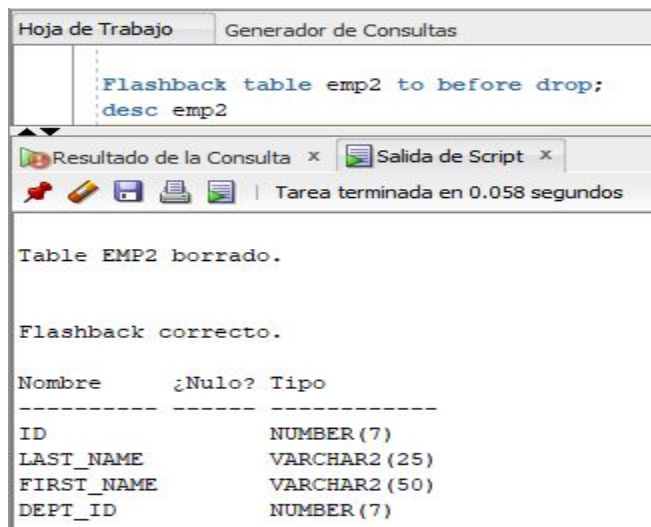
8. Consulte la papelera de reciclaje para comprobar si está la tabla.



The screenshot shows the SQL Developer interface. The top pane contains the SQL query: `SELECT * FROM recyclebin;`. The bottom pane shows the results of the query. The status bar indicates 'Todas las Filas Recuperadas: 2 en 0.012 segundos'. The results are displayed in a table with three columns: OBJECT_NAME, ORIGINAL_NAME, and OPERATION.

	OBJECT_NAME	ORIGINAL_NAME	OPERATION
1	BIN\$8SSvuN3XS+WJWk0Uz1R2aQ==\$0	EMP2	DROP

9. Anule el borrado de la tabla EMP2.



The screenshot shows the SQL Developer interface. The top pane contains the SQL query: `Flashback table emp2 to before drop; desc emp2`. The bottom pane shows the results of the query. The status bar indicates 'Tarea terminada en 0.058 segundos'. The results are displayed in a table with three columns: Nombre, ¿Nulo?, and Tipo.

Nombre	¿Nulo?	Tipo
ID		NUMBER (7)
LAST_NAME		VARCHAR2 (25)
FIRST_NAME		VARCHAR2 (50)
DEPT_ID		NUMBER (7)

10. Borre la columna FIRST_NAME de la tabla EMPLOYEES2. Confirme la modificación comprobando la descripción de la tabla.

```
alter table Employees2 drop column FIRST_NAME;
desc employees2
```

Resultado de la Consulta x Salida de Script x

Tarea terminada en 0.248 segundos

Table EMPLOYEES2 alterado.

Nombre	¿Nulo?	Tipo
ID		NUMBER(6)
LAST_NAME	NOT NULL	VARCHAR2(25)
SALARY		NUMBER(8,2)
DEPT_ID		NUMBER(4)

11. En la tabla EMPLOYEES2, marque la columna DEPT_ID como UNUSED. Confirme la modificación comprobando la descripción de la tabla.

```
alter table employees2 set unused (dept_id)
desc employees2
```

Resultado de la Consulta x Salida de Script x

Tarea terminada en 0.059 segundos

Table EMPLOYEES2 alterado.

Nombre	¿Nulo?	Tipo
ID		NUMBER(6)
LAST_NAME	NOT NULL	VARCHAR2(25)
SALARY		NUMBER(8,2)

12. Borre todas las columnas UNUSED de la tabla EMPLOYEES2. Confirme la modificación comprobando la descripción de la tabla.

```
ALTER TABLE employees2
DROP UNUSED COLUMNS;
desc employees2
```

Resultado de la Consulta x Salida de Script x

Tarea terminada en 0.078 seg

Table EMPLOYEES2 alterado.

Table EMPLOYEES2 alterado.

Nombre	¿Nulo?	Tipo
ID		NUMBER(6)
LAST_NAME	NOT NULL	VARCHAR2(25)
SALARY		NUMBER(8,2)

13. Agregue una restricción PRIMARY KEY de nivel de tabla a la tabla EMP2 en la columna ID. Se debe asignar un nombre a la restricción en el momento de la creación. Asigne a la restricción el nombre my_emp_id_pk.

```
alter table emp2 add constraint my_emp_id_pk primary key (id);
desc emp2
```

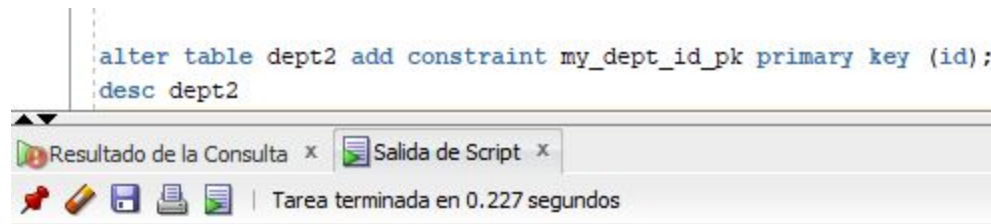
Resultado de la Consulta x Salida de Script x

Tarea terminada en 0.061 segundos

Nombre	¿Nulo?	Tipo
ID	NOT NULL	NUMBER(7)
LAST_NAME		VARCHAR2(25)
FIRST_NAME		VARCHAR2(50)
DEPT_ID		NUMBER(7)

14. Cree una restricción PRIMARY KEY a la tabla DEPT2 mediante la columna ID. Se debe asignar un nombre a la restricción en el momento de la creación. Asigne a la restricción el nombre my_dept_id_pk.

```
alter table dept2 add constraint my_dept_id_pk primary key (id);
desc dept2
```



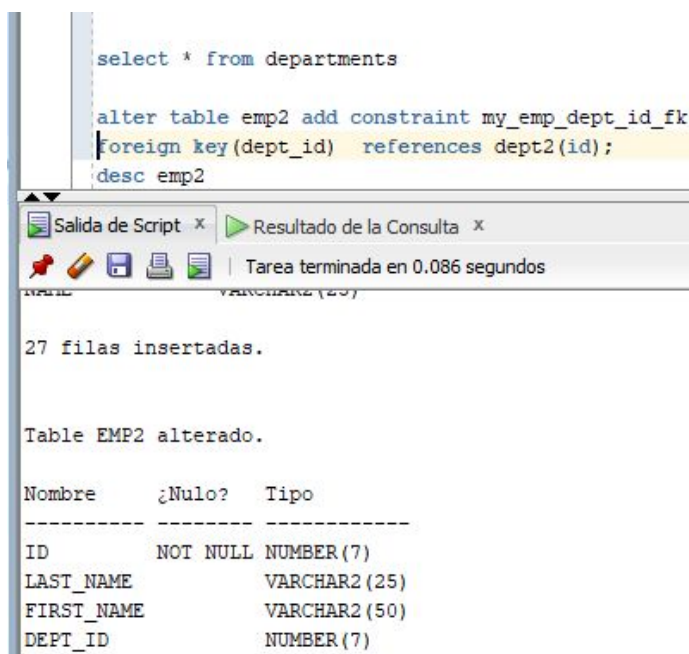
Nombre	¿Nulo?	Tipo

ID	NOT NULL	NUMBER(7)
NAME		VARCHAR2(25)

15. Agregue una referencia de clave ajena en la tabla EMP2 que asegure que el empleado no está asignado a un departamento inexistente. Asigne a la restricción el nombre my_emp_dept_id_fk.

```
select * from departments

alter table emp2 add constraint my_emp_dept_id_fk
foreign key(dept_id) references dept2(id);
desc emp2
```



27 filas insertadas.

Table EMP2 alterado.

Nombre	¿Nulo?	Tipo

ID	NOT NULL	NUMBER(7)
LAST_NAME		VARCHAR2(25)
FIRST_NAME		VARCHAR2(50)
DEPT_ID		NUMBER(7)

16. Confirme que las restricciones se han agregado consultando la vista USER_CONSTRAINTS. Tenga en cuenta los tipos y los nombres de las restricciones.

```
select CONSTRAINT_NAME, CONSTRAINT_TYPE
from user_constraints
where table_name= 'EMP2' or table_name='DEPT2';
```

	CONSTRAINT_NAME	CONSTRAINT_TYPE
1	MY_DEPT_ID_PK	P
2	MY_EMP_ID_PK	P
3	MY_EMP_DEPT_ID_FK	R

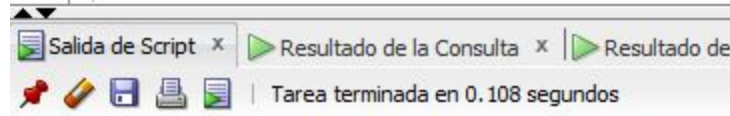
17. Muestre los tipos y los nombres de objeto de la vista del diccionario de datos USER_OBJECT para las tablas EMP2 y DEPT2. Observe que se han creado tablas nuevas y un nuevo índice.

```
SELECT object_name, object_type FROM user_objects
where object id between 21164 and 21251
```

	OBJECT_NAME	OBJECT_TYPE
1	MY_DEPT_ID_PK	INDEX
2	DEPT2	TABLE
3	MY_EMP_ID_PK	INDEX
4	EMP2	TABLE

18. Modifique la tabla EMP2. Agregue una columna COMMISSION de tipo de datos NUMBER, precisión 2, escala 2. Agregue una restricción a la columna COMMISSION que garantice que el valor de comisión es mayor que cero.

```
alter table emp2 add ( COMMISSION NUMBER (2,2)
constraint com_check check (comission >0));
DESC EMP2
```



Salida de Script x Resultado de la Consulta x Resultado de

Tarea terminada en 0.108 segundos

Table EMP2 alterado.

Nombre	¿Nulo?	Tipo
-----	-----	-----
ID	NOT NULL	NUMBER (7)
LAST_NAME		VARCHAR2 (25)
FIRST_NAME		VARCHAR2 (50)
DEPT_ID		NUMBER (7)
COMMISSION		NUMBER (2,2)

19. Borre las tablas EMP2 y DEPT2 de forma que no se puedan restaurar. Verifique la papelera de reciclaje.



Hoja de Trabajo Generador de Consultas

```
--19
DROP TABLE EMP2 PURGE;
DROP TABLE DEPT2 PURGE;
SELECT * FROM RECYCLEBIN WHERE ORIGINAL_NAME = 'EMP2' OR ORIGINAL_NAME = 'DEPT2';
```

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 0 en 0.009 segundos

OBJECT...	ORIGINAL...	OPERATION	TYPE	TS_NAME	CREATETI...	DROPTIME	DROPSN	PARTITIO...	CAN_UND...	CAN_PURGE	RELATED	BASE_OB...	PURGE
-----------	-------------	-----------	------	---------	-------------	----------	--------	-------------	------------	-----------	---------	------------	-------

20. Cree la tabla DEPT_NAMED_INDEX basada en el siguiente gráfico de instancia de tabla. Asigne al índice de la columna PRIMARY KEY el nombre DEPT_PK_IDX.

Column Name	Deptno	Dname
Primary Key	Yes	
Data type	NUMBER	VARCHAR2
Length	4	30

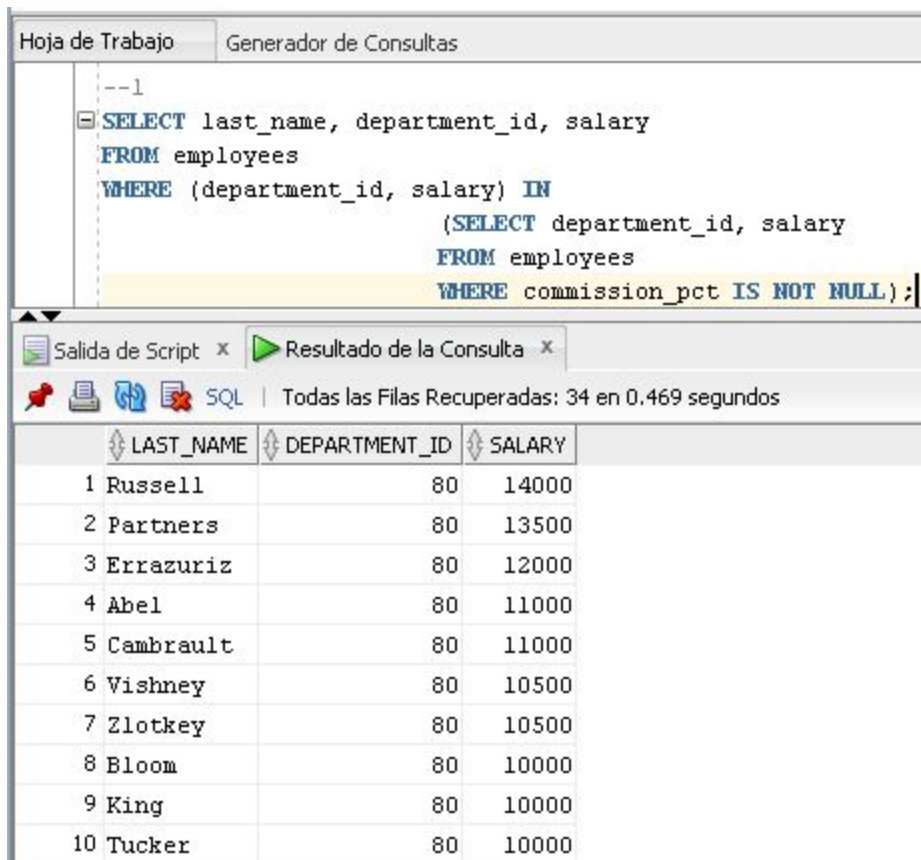
The screenshot shows a database IDE with two main windows. The top window, titled 'Hoja de Trabajo' and 'Generador de Consultas', contains the following SQL script:

```
--20
CREATE TABLE DEPT_NAMED_INDEX (
  Deptno NUMBER(4) PRIMARY KEY USING INDEX(CREATE INDEX DEPT_PK_IDX ON DEPT_NAMED_INDEX(Deptno)),
  Dname VARCHAR2(30));
```

The bottom window, titled 'Salida de Script', shows the execution result: 'Tarea terminada en 0.078 segundos' and 'Table DEPT_NAMED_INDEX creado.'

Realice los siguientes ejercicios luego de estudiar el documento Les06-Subqueries.ppt:

1. Escriba una consulta para mostrar el apellido, el número de departamento y el salario de cualquier empleado cuyo número de departamento y salario se correspondan con el número de departamento y el salario de cualquier empleado que gane una comisión.



The screenshot shows a SQL IDE interface with two tabs: 'Hoja de Trabajo' and 'Generador de Consultas'. The 'Generador de Consultas' tab is active, displaying the following SQL query:

```
--1
SELECT last_name, department_id, salary
FROM employees
WHERE (department_id, salary) IN
      (SELECT department_id, salary
       FROM employees
       WHERE commission_pct IS NOT NULL);
```

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 status bar indicates 'Todas las Filas Recuperadas: 34 en 0.469 segundos'.

	LAST_NAME	DEPARTMENT_ID	SALARY
1	Russell	80	14000
2	Partners	80	13500
3	Errazuriz	80	12000
4	Abel	80	11000
5	Cambrault	80	11000
6	Vishney	80	10500
7	Zlotkey	80	10500
8	Bloom	80	10000
9	King	80	10000
10	Tucker	80	10000

2. Muestre el apellido, el nombre de departamento y el salario de cualquier empleado cuyo salario y comisión se correspondan con el salario y la comisión de cualquier empleado con el identificador de ubicación 1700.

Hoja de Trabajo Generador de Consultas

```
--2
SELECT e.last_name, d.department_name, e.salary
FROM employees e JOIN departments d
ON (d.department_id = e.department_id)
WHERE (e.salary, NVL(e.commission_pct,0)) IN
      (SELECT e.salary, NVL(e.commission_pct,0)
       FROM employees e JOIN departments d
       ON (e.department_id = d.department_id)
       WHERE d.location_id = 1700);
```

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 36 en 0 segundos

	LAST_NAME	DEPARTMENT_NAME	SALARY
1	King	Executive	24000
2	De Haan	Executive	17000
3	Kochhar	Executive	17000
4	Higgins	Accounting	12008
5	Greenberg	Finance	12008
6	Faviet	Finance	9000
7	Hunold	IT	9000
8	Chen	Finance	8200
9	Fripp	Shipping	8200
10	Sciarra	Finance	7700

3. Muestre los detalles del identificador de empleado, el apellido y el identificador de departamento de los empleados que vivan en ciudades cuyo nombre empiece por T.

Hoja de Trabajo Generador de Consultas

```
--3
SELECT e.employee_id, e.last_name, e.department_id
FROM employees e JOIN departments d
ON (e.department_id = d.department_id)
WHERE d.location_id IN (SELECT location_id
                        FROM locations WHERE
                        city LIKE 'T%');
```

Salida de Script x Resultado de la Consulta x

SQL | Todas las Filas Recuperadas: 2 en 0 segundos

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
1	201 Hartstein	20
2	202 Fay	20

4. Escriba una consulta para buscar todos los empleados que ganen más que el salario medio de su departamento. Muestre el apellido, el salario, el identificador de departamento y el salario medio del departamento. Ordene por salario medio. Utilice alias para las columnas recuperadas por la consulta como se muestra en el ejemplo de salida.

Hoja de Trabajo Generador de Consultas

--4

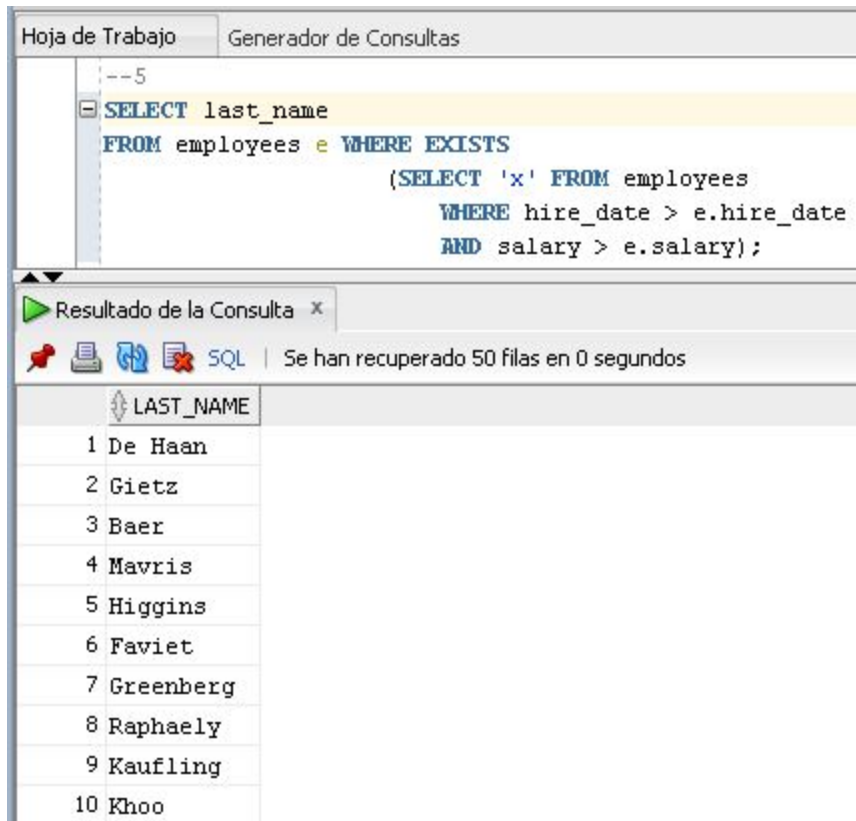
```
SELECT last_name, department_id, salary, (SELECT ROUND(AVG(salary))
      FROM employees
      WHERE department_id = e.department_id) AS PROMEDIO
FROM employees e
WHERE e.salary > (SELECT AVG(salary)
      FROM employees
      WHERE department_id = e.department_id)
ORDER BY PROMEDIO;
```

Resultado de la Consulta x

Todas las Filas Recuperadas: 38 en 0.016 segundos

	LAST_NAME	DEPARTMENT_ID	SALARY	PROMEDIO
1	Dilly	50	3600	3476
2	Chung	50	3800	3476
3	Bell	50	4000	3476
4	Weiss	50	8000	3476
5	Fripp	50	8200	3476
6	Kaufling	50	7900	3476
7	Bull	50	4100	3476
8	Sarchand	50	4200	3476
9	Everett	50	3900	3476
10	Rajs	50	3500	3476

5. Escriba una consulta para mostrar los apellidos de los empleados que tienen uno o más colegas en su departamento con fechas de contratación posteriores pero salarios más altos.



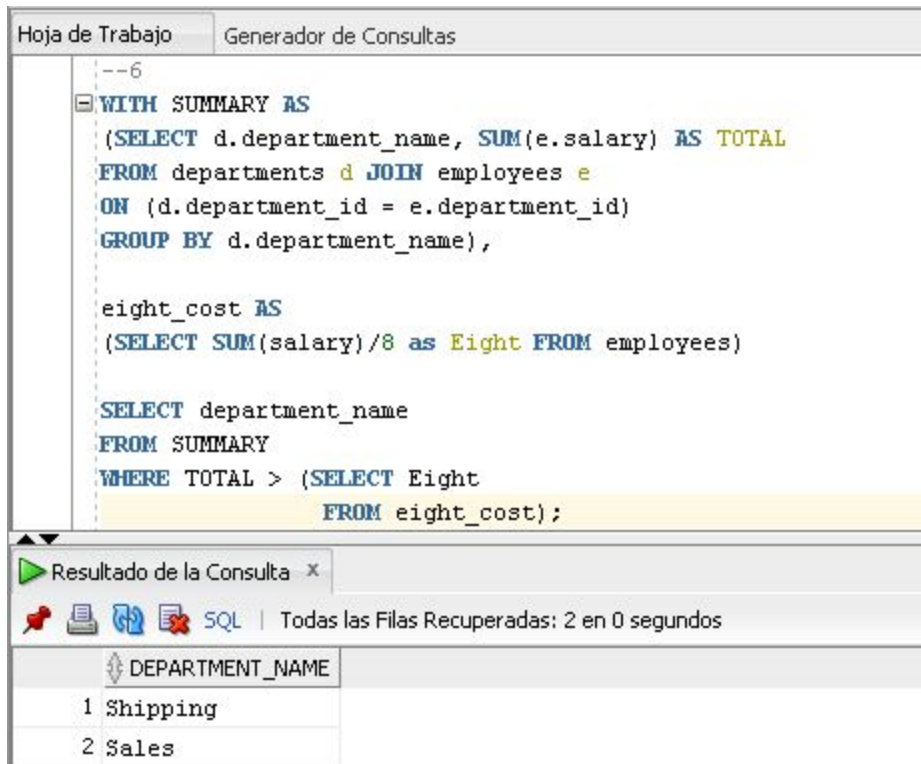
The screenshot shows a SQL IDE with two tabs: 'Hoja de Trabajo' and 'Generador de Consultas'. The 'Generador de Consultas' tab is active, displaying the following SQL query:

```
--5
SELECT last_name
FROM employees e WHERE EXISTS
    (SELECT 'x' FROM employees
     WHERE hire_date > e.hire_date
     AND salary > e.salary);
```

Below the query editor, there is a tab titled 'Resultado de la Consulta'. It shows a status bar indicating 'Se han recuperado 50 filas en 0 segundos'. The results are displayed in a table with the column 'LAST_NAME'.

	LAST_NAME
1	De Haan
2	Gietz
3	Baer
4	Mavris
5	Higgins
6	Faviet
7	Greenberg
8	Raphaely
9	Kaufling
10	Khoo

6. Escriba una consulta para mostrar los nombres de departamento de los departamentos cuyo costo de salario total supere un octavo ($1/8$) del costo de salario total de toda la compañía. Utilice la cláusula WITH para escribir esta consulta. Asigne a la consulta el nombre SUMMARY.



The screenshot shows a SQL query editor with two tabs: "Hoja de Trabajo" and "Generador de Consultas". The query is as follows:

```
--6
WITH SUMMARY AS
  (SELECT d.department_name, SUM(e.salary) AS TOTAL
   FROM departments d JOIN employees e
   ON (d.department_id = e.department_id)
   GROUP BY d.department_name),

  eight_cost AS
  (SELECT SUM(salary)/8 as Eight FROM employees)

SELECT department_name
FROM SUMMARY
WHERE TOTAL > (SELECT Eight
               FROM eight_cost);
```

Below the query editor, there is a section titled "Resultado de la Consulta" with a status bar indicating "Todas las Filas Recuperadas: 2 en 0 segundos". The results are displayed in a table with the following data:

	DEPARTMENT_NAME
1	Shipping
2	Sales