



Bases de datos

Actividades del tema 6

YEHOR BURLACHENKO



Prueba

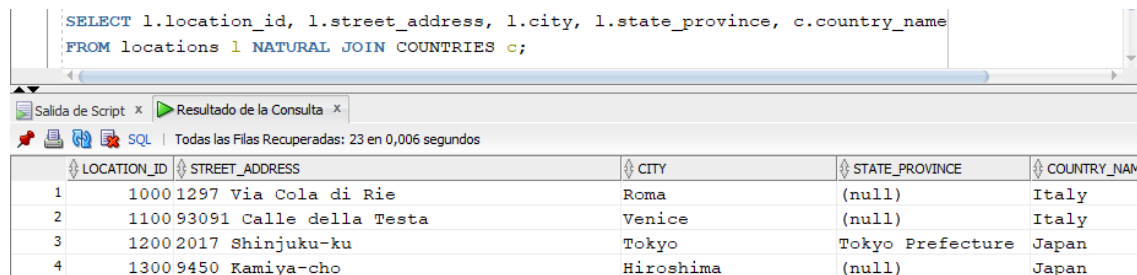
Los tipos de unión no soportados por Oracle son:

- Uniones no igualitarias
- La opción de OUTER en la Unión OUTER izquierda y Unión OUTER derecha si se refiere explícitamente al uso de OUTER.

Actividades.

Ejercicio 1. Escriba una consulta para que el departamento de recursos humanos genere las direcciones de todos los departamentos. Utilice las tablas LOCATIONS y COUNTRIES. Muestre el ID de ubicación, dirección, ciudad, estado o provincia y país en la salida. Utilice NATURAL JOIN para producir los resultados.

Respuesta:



The screenshot shows an SQL query execution window. The query is: `SELECT l.location_id, l.street_address, l.city, l.state_province, c.country_name FROM locations l NATURAL JOIN COUNTRIES c;`. The results are displayed in a table with 5 columns: LOCATION_ID, STREET_ADDRESS, CITY, STATE_PROVINCE, and COUNTRY_NAME. There are 4 rows of data.

LOCATION_ID	STREET_ADDRESS	CITY	STATE_PROVINCE	COUNTRY_NAME
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

Ejercicio 2. El departamento de recursos humanos necesita un informe sólo de los empleados con los departamentos correspondientes. Escriba una consulta para mostrar el apellido, número y nombre de departamento de estos empleados.

Respuesta:

```
-- 2
SELECT e.last_name, e.department_id, d.department_name
FROM employees e JOIN departments d
ON(e.department_id = d.department_id);
```

	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1	Whalen	10	Administration
2	Fay	20	Marketing
3	Hartstein	20	Marketing
4	Tobias	30	Purchasing
5	Colmenares	30	Purchasing
6	Baida	30	Purchasing
7	Raphaely	30	Purchasing
8	Khoo	30	Purchasing
9	Himuro	30	Purchasing

Ejercicio 3. El departamento de recursos humanos necesita un informe de todos los empleados de Toronto. Muestre el apellido, cargo, número y nombre de departamento de todos los empleados que trabajan en Toronto.

Respuesta:

```
-- 3
SELECT e.last_name, e.job_id, e.department_id, d.department_name
FROM employees e
JOIN departments d
ON(e.department_id = d.department_id)
JOIN locations l
ON(l.location_id = d.location_id)
WHERE l.city LIKE 'Toronto';
```

	LAST_NAME	JOB_ID	DEPARTMENT_ID	DEPARTMENT_NAME
1	Hartstein	MK_MAN	20	Marketing
2	Fay	MK_REP	20	Marketing

Ejercicio 4. Cree un informe para mostrar el apellido y número de empleado junto con el apellido y número de gestor del empleado, Etiquete las columnas como Employee, Emp#. Manager y Mgr#, respectivamente.

Respuesta:

```

SELECT e.last_name AS "Employee", e.employee_id AS "emp#",
       m.last_name AS "Manager", m.employee_id AS "Mng#"
FROM employees e JOIN employees m
ON e.manager_id = m.employee_id;

```

	Employee	emp#	Manager	Mng#
1	Kumar	173	Cambrault	148
2	Bates	172	Cambrault	148
3	Smith	171	Cambrault	148
4	Fox	170	Cambrault	148
5	Bloom	169	Cambrault	148
6	Ozer	168	Cambrault	148
7	Hunold	103	De Haan	102
8	Banda	167	Errazuriz	147

Ejercicio 5. Modifique lab_06_04. sql para mostrar todos los empleados, incluido King, que no tienen gestor. Ordene los resultados por número de empleado. Guarde la sentencia SQL como lab_06_05. sql. Ejecute la consulta en el archivo lab_06_05.sql.

Respuesta:

```

SELECT e.last_name AS "Employee", e.employee_id AS "emp#",
       m.last_name AS "Manager", m.employee_id AS "Mng#"
FROM employees e LEFT OUTER JOIN employees m
ON e.manager_id = m.employee_id;

```

	Employee	emp#	Manager	Mng#
1	Kumar	173	Cambrault	148
2	Bates	172	Cambrault	148
3	Smith	171	Cambrault	148
4	Fox	170	Cambrault	148
5	Bloom	169	Cambrault	148
6	Ozer	168	Cambrault	148
7	Hunold	103	De Haan	102

Ejercicio 6. Cree un informe del departamento de recursos humanos que muestre los apellidos y números de empleado y todos los empleados que trabajen en el mismo departamento como un empleado determinado. Proporcione a cada columna una etiqueta adecuada. Guarde el script en un archivo denominado lab_06_06. sql.

Respuesta:

```

SELECT e.department_id, e.last_name AS "Employee", c.last_name AS "Colleague"
FROM employees e
INNER JOIN employees c
ON e.department_id = c.department_id
AND e.employee_id <> c.employee_id;

```

	DEPARTMENT_ID	Employee	Colleague
1	90	De Haan	King
2	90	Kochhar	King
3	90	De Haan	Kochhar
4	90	King	Kochhar
5	90	Kochhar	De Haan
6	90	King	De Haan

Ejercicio 8. El departamento de recursos humanos desea determinar los nombres de todos los empleados contratados después de Davies. Cree una consulta para mostrar el nombre y la fecha de contratación de cualquier empleado contratado después del empleado Davies.

Respuesta:

```

SELECT e.last_name, e.hire_date
FROM employees e JOIN employees d
ON (e.hire_date > d.hire_date)
WHERE d.last_name like 'Davies';

```

	LAST_NAME	HIRE_DATE
1	Kochhar	21/09/05
2	Hunold	03/01/06
3	Ernst	21/05/07
4	Austin	25/06/05

Ejercicio 9. El departamento de recursos humanos necesita buscar el nombre y la fecha de contratación de todos los empleados contratados antes que sus gestores, junto con el denominado lab_06_09.sql.

Respuesta:

```

SELECT e.last_name, e.hire_date, m.last_name, m.hire_date
FROM employees e JOIN employees m
ON (e.manager_id = m.employee_id)
WHERE e.hire_date < m.hire_date;

```

	LAST_NAME	HIRE_DATE	LAST_NAME_1	HIRE_DATE_1
1	Kaufling	01/05/03	King	17/06/03
2	Raphaely	07/12/02	King	17/06/03
3	De Haan	13/01/01	King	17/06/03
4	Higgins	07/06/02	Kochhar	21/09/05
5	Baer	07/06/02	Kochhar	21/09/05