Oracle LAB 6

Exercise 1:

The HR department needs a report of all employees. Write a query to display the last name, department number, and department name for all employees.

The Result:

```
SELECT e.last_name, d.department_id, d.department_name
FROM employees e
JOIN departments d
ON (e.department_id = d.department_id)
ORDER BY e.department_id;
```
Exercise 2:

A) Create a report to display employees’ last name and employee number along with their manager’s last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively.

B) Modify Part A to display all employees including King, who has no manager. Order the results by the employee number.

The Result:
Exercise 3:

The HR department needs to find the names and hire dates for all employees who were hired before their managers, along with their managers’ names and hire dates.

The Result:

```
SELECT e.last_name 'Employee', e.hire_date "Emp Hire Date", m.last_name 'Manager', m.hire_date "Mgr Hire Date"
FROM employees e
JOIN employees m
ON (e.manager_id = m.employee_id)
WHERE e.hire_date < m.hire_date;
```
Exercise 4:

Display the employee number, last name, and salary of all employees who earn more than the average salary and who work in a department with any employee whose last name contains a 'u'.

The Result:

```sql
SELECT employee_id, last_name, salary
FROM employees
WHERE department_id IN (SELECT department_id FROM employees WHERE last_name LIKE '%u%')
AND salary > (SELECT AVG(salary) FROM employees);
```
Exercise 5:

The HR department needs a report with the following specifications:
- Last name and department ID of all the employees from the EMPLOYEES table, regardless of whether or not they belong to a department
- Department ID and department name of all the departments from the DEPARTMENTS table, regardless of whether or not they have employees working in them

Write a compound query to accomplish this.

The Result:

```sql
SELECT last_name, department_id, TO_CHAR(NULL)
FROM employees
UNION

SELECT TO_CHAR(NULL), department_id, department_name
FROM departments;
```
Exercise 6:

Create a report that lists the employee IDs and job IDs of those employees who currently have a job title that is the same as their job title when they were initially hired by the company (that is, they changed jobs but have now gone back to doing their original job).

The Result:

```
SELECT employees_id, job_id
FROM employees
INTERSECT
SELECT employees_id, job_id
FROM job_history;
```
Exercise 7:

The HR department needs a list of countries that have no departments located in them. Display the country ID and the name of the countries. Use set operators to create this report.

The Result:

```sql
SELECT country_id, country_name
FROM countries
MINUS
SELECT l.country_id, c.country_name
FROM locations l JOIN countries c
ON (l.country_id = c.country_id);
```

<table>
<thead>
<tr>
<th>COUNTRY_ID</th>
<th>COUNTRY_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Argentina</td>
</tr>
<tr>
<td>2</td>
<td>Belgium</td>
</tr>
<tr>
<td>3</td>
<td>Denmark</td>
</tr>
<tr>
<td>4</td>
<td>Egypt</td>
</tr>
<tr>
<td>5</td>
<td>France</td>
</tr>
<tr>
<td>6</td>
<td>HongKong</td>
</tr>
<tr>
<td>7</td>
<td>Israel</td>
</tr>
<tr>
<td>8</td>
<td>Kuwait</td>
</tr>
<tr>
<td>9</td>
<td>Nigeria</td>
</tr>
<tr>
<td>10</td>
<td>Zambia</td>
</tr>
</tbody>
</table>

6.34 10 rows selected in 0.157 seconds (more...)