

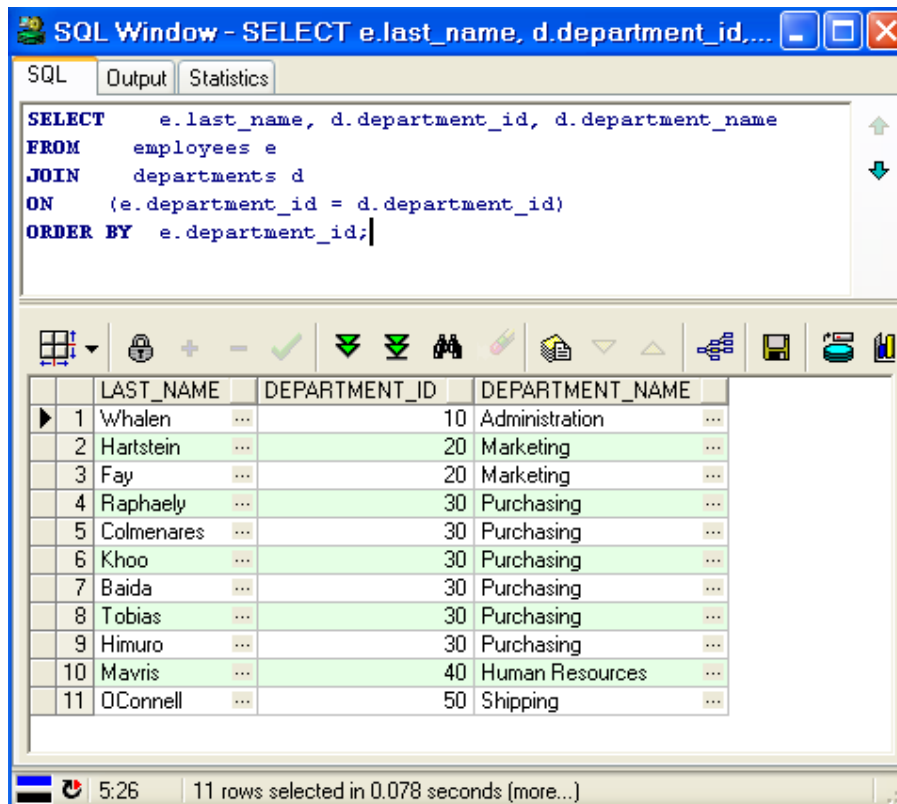


Oracle LAB 6

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:



The screenshot shows an Oracle SQL Window titled "SQL Window - SELECT e.last_name, d.department_id,...". The window has three tabs: "SQL", "Output", and "Statistics". The "SQL" tab is active, displaying the following query:

```
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;
```

Below the query is a toolbar with various icons. The "Output" tab is selected, showing a table with 11 rows of data. The table has four columns: "LAST_NAME", "DEPARTMENT_ID", and "DEPARTMENT_NAME". The data is as follows:

| | LAST_NAME | DEPARTMENT_ID | DEPARTMENT_NAME |
|----|------------|---------------|-----------------|
| 1 | Whalen | 10 | Administration |
| 2 | Hartstein | 20 | Marketing |
| 3 | Fay | 20 | Marketing |
| 4 | Raphaely | 30 | Purchasing |
| 5 | Colmenares | 30 | Purchasing |
| 6 | Khoo | 30 | Purchasing |
| 7 | Baida | 30 | Purchasing |
| 8 | Tobias | 30 | Purchasing |
| 9 | Himuro | 30 | Purchasing |
| 10 | Mavris | 40 | Human Resources |
| 11 | OConnell | 50 | Shipping |

At the bottom of the window, a status bar shows the execution time as 5:26 and the message "11 rows selected in 0.078 seconds (more...)".



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:

SQL Window - ex2 a.sql

```
SELECT e.last_name "Employee", e.employee_id "Emp#",  
       m.last_name "Manager", m.employee_id "Mgr#"   
FROM   employees e JOIN employees m   
ON     (e.manager_id = m.employee_id);
```

| | Employee | Emp# | Manager | Mgr# |
|----|-----------|------|-----------|------|
| 1 | OConnell | 198 | Mourgos | 124 |
| 2 | Grant | 199 | Mourgos | 124 |
| 3 | Whalen | 200 | Kochhar | 101 |
| 4 | Hartstein | 201 | King | 100 |
| 5 | Fay | 202 | Hartstein | 201 |
| 6 | Mavris | 203 | Kochhar | 101 |
| 7 | Baer | 204 | Kochhar | 101 |
| 8 | Higgins | 205 | Kochhar | 101 |
| 9 | Gietz | 206 | Higgins | 205 |
| 10 | Kochhar | 101 | King | 100 |
| 11 | De Haan | 102 | King | 100 |
| 12 | Hunold | 103 | De Haan | 102 |
| 13 | Ernst | 104 | Hunold | 103 |

SQL Window - ex 2 b.sql

```
SELECT e.last_name "Employee", e.employee_id "Emp#",  
       m.last_name "Manager", m.employee_id "Mgr#"   
FROM   employees e   
LEFT OUTER JOIN employees m   
ON     (e.manager_id = m.employee_id)   
ORDER BY e.employee_id;
```

| | Employee | Emp# | Manager | Mgr# |
|----|-----------|------|-----------|------|
| 1 | King | 100 | | |
| 2 | Kochhar | 101 | King | 100 |
| 3 | De Haan | 102 | King | 100 |
| 4 | Hunold | 103 | De Haan | 102 |
| 5 | Ernst | 104 | Hunold | 103 |
| 6 | Austin | 105 | Hunold | 103 |
| 7 | Pataballa | 106 | Hunold | 103 |
| 8 | Lorentz | 107 | Hunold | 103 |
| 9 | Greenberg | 108 | Kochhar | 101 |
| 10 | Faviet | 109 | Greenberg | 108 |
| 11 | Chen | 110 | Greenberg | 108 |

SQL script saved successfully



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:

The screenshot shows a SQL window titled "SQL Window - ex 3.sql". The query is as follows:

```
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 ;
```

The results are displayed in a table with the following columns: Employee, Emp Hire Date, Manager, and Mgr HIRE DATE. The table contains 11 rows of data.

| | Employee | Emp Hire Date | Manager | Mgr HIRE DATE |
|----|----------|---------------|-----------|---------------|
| 1 | OConnell | 21/06/1999 | Mourgos | 16/11/1999 |
| 2 | Whalen | 17/09/1987 | Kochhar | 21/09/1989 |
| 3 | Hunold | 03/01/1990 | De Haan | 13/01/1993 |
| 4 | Faviet | 16/08/1994 | Greenberg | 17/08/1994 |
| 5 | Marlow | 16/02/1997 | Fripp | 10/04/1997 |
| 6 | Ladwig | 14/07/1995 | Vollman | 10/10/1997 |
| 7 | Rajs | 17/10/1995 | Mourgos | 16/11/1999 |
| 8 | Davies | 29/01/1997 | Mourgos | 16/11/1999 |
| 9 | Matos | 15/03/1998 | Mourgos | 16/11/1999 |
| 10 | Vargas | 09/07/1998 | Mourgos | 16/11/1999 |
| 11 | King | 30/01/1996 | Partners | 05/01/1997 |

The status bar at the bottom indicates "2:1 SQL script saved successfully".



Islamic University of Gaza
Faculty of Engineering
Computer Department

Advanced DataBase ECOM 5054
Instructor: Eng. Husam Alzaq
T.A: Eng. Doaa KH. Abu Jabal

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 Window - SELECT employee_id, last_name, salary FROM employees WHERE ...

SQL Output Statistics

```
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);
```

| | EMPLOYEE_ID | LAST_NAME | SALARY |
|----|-------------|------------|----------|
| 1 | 103 | Hunold | 9000.00 |
| 2 | 114 | Raphaely | 11000.00 |
| 3 | 123 | Vollman | 6500.00 |
| 4 | 122 | Kaufling | 7900.00 |
| 5 | 121 | Fripp | 8200.00 |
| 6 | 120 | Weiss | 8000.00 |
| 7 | 177 | Livingston | 8400.00 |
| 8 | 176 | Taylor | 8600.00 |
| 9 | 175 | Hutton | 8800.00 |
| 10 | 174 | Abel | 11000.00 |
| 11 | 172 | Bates | 7300.00 |
| 12 | 171 | Smith | 7400.00 |

4:50 12 rows selected in 0.046 seconds (more...)



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:

The screenshot shows a SQL window titled "SQL Window - ex 5.sql" with three tabs: SQL, Output, and Statistics. The SQL tab is active, displaying the following query:

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

UNION

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

Below the query, the results are displayed in a table with four columns: LAST_NAME, DEPARTMENT_ID, and TO_CHAR(NULL). The table contains 9 rows of data, with the first two columns highlighted in green and the third column highlighted in yellow.

| | LAST_NAME | DEPARTMENT_ID | TO_CHAR(NULL) |
|---|-----------|---------------|---------------|
| 1 | Abel | 80 | ... |
| 2 | Ande | 80 | ... |
| 3 | Atkinson | 50 | ... |
| 4 | Austin | 60 | ... |
| 5 | Baer | 70 | ... |
| 6 | Baida | 30 | ... |
| 7 | Banda | 80 | ... |
| 8 | Bates | 80 | ... |
| 9 | Bell | 50 | ... |

The status bar at the bottom indicates "7:18" and "SQL script saved successfully".



Islamic University of Gaza
Faculty of Engineering
Computer Department

Advanced DataBase ECOM 5054
Instructor: Eng. Husam Alzaq
T.A: Eng. Doaa KH. Abu Jabal

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:

The screenshot shows an SQL Window titled "SQL Window - ex 6.sql". It has three tabs: "SQL", "Output", and "Statistics". The "SQL" tab is active, displaying the following query:

```
SELECT employee_id, job_id
FROM employees

INTERSECT

SELECT employee_id, job_id
FROM job_history;
```

Below the query editor is a toolbar with various icons. Below the toolbar is a table with the following data:

| | EMPLOYEE_ID | JOB_ID |
|---|-------------|---------|
| 1 | 176 | SA_REP |
| 2 | 200 | AD_ASST |

At the bottom of the window, there is a status bar showing the time "7:18" and the message "SQL script saved successfully".



Islamic University of Gaza
Faculty of Engineering
Computer Department

Advanced DataBase ECOM 5054
Instructor: Eng. Husam Alzaq
T.A: Eng. Doaa KH. Abu Jabal

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:

The screenshot shows a SQL Window titled "SQL Window - SELECT country_...". The window has three tabs: "SQL", "Output", and "Statistics". The "SQL" tab is active, displaying the following query:

```
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);
```

Below the query editor is a toolbar with various icons. The results are displayed in a table with two columns: "COUNTRY_ID" and "COUNTRY_NAME". The table contains 10 rows of data, with the first row highlighted in blue and the rest in green.

| | COUNTRY_ID | COUNTRY_NAME |
|----|------------|--------------|
| 1 | AR | Argentina |
| 2 | BE | Belgium |
| 3 | DK | Denmark |
| 4 | EG | Egypt |
| 5 | FR | France |
| 6 | HK | HongKong |
| 7 | IL | Israel |
| 8 | KW | Kuwait |
| 9 | NG | Nigeria |
| 10 | ZM | Zambia |

At the bottom of the window, a status bar shows the time "6:34" and the message "10 rows selected in 0.157 seconds (more...)".