Practice 3-1: Restricting and Sorting Data

Overview

In this practice, you build more reports by using statements that use the WHERE clause and the ORDER BY clause. You make the SQL statements more reusable and generic by including the ampersand substitution.

Task

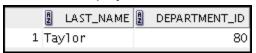
The HR department needs your assistance in creating some queries.

1. Because of budget issues, the HR department needs a report that displays the last name and salary of employees who earn more than \$12,000. Save your SQL statement as a file named lab 03 01.sql. Run your query.

	LAST_NAME	2 SALARY
1	King	24000
2	Kochhar	17000
3	De Haan	17000
4	Hartstein	13000
5	Higgins	12008

2. Open a new SQL Worksheet. Create a report that displays the last name and department number for employee number 176. Run the query.

Oracle University and Egabi Solutions use only



3. The HR department needs to find high-salary and low-salary employees. Modify lab_03_01.sql to display the last name and salary for any employee whose salary is not in the range \$5,000 through \$12,000. Save your SQL statement as lab_03_03.sql.



4. Create a report to display the last name, job ID, and hire date for employees with the last names of Matos and Taylor. Order the query in ascending order by hire date.



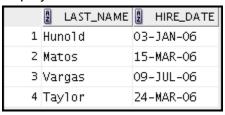
5. Display the last name and department ID of all employees in departments 20 or 50 in ascending alphabetical order by name.

	aning dipridabotic	· · · · · · · · · · · · · · · · · · ·
	LAST_NAME	DEPARTMENT_ID
1	Davies	50
2	Fay	20
3	Hartstein	20
4	Matos	50
5	Mourgos	50
6	Rajs	50
7	Vargas	50

6. Modify lab_03_03.sql to display the last name and salary of employees who earn between \$5,000 and \$12,000, and are in department 20 or 50. Label the columns Employee and Monthly Salary, respectively. Save lab_03_03.sql as lab_03_06.sql again. Run the statement in lab_03_06.sql.

	Employee	A	Monthly Salary
1	Fay		6000
2	Mourgos		5800

7. The HR department needs a report that displays the last name and hire date of all employees who were hired in 2006.



8. Create a report to display the last name and job title of all employees who do not have a manager.



9. Create a report to display the last name, salary, and commission of all employees who earn commissions. Sort the data in descending order of salary and commissions.

Use the column's numeric position in the ORDER BY clause.

	LAST_NAME	2 SALARY	② COMMISSION_PCT
1	Abe1	11000	0.3
2	Zlotkey	10500	0.2
3	Taylor	8600	0.2
4	Grant	7000	0.15

10. Members of the HR department want to have more flexibility with the queries that you are writing. They would like a report that displays the last name and salary of employees who earn more than an amount that the user specifies after a prompt. Save this query to a file named lab_03_10.sql. (You can use the query created in Task 1 and modify it.) If you enter 12000 when prompted, the report displays the following results:

	LAST_NAME	2 SALARY
1	King	24000
2	Kochhar	17000
3	De Haan	17000
4	Hartstein	13000
5	Higgins	12008

11. The HR department wants to run reports based on a manager. Create a query that prompts the user for a manager ID, and generates the employee ID, last name, salary, and department for that manager's employees. The HR department wants the ability to sort the report on a selected column. You can test the data with the following values:

manager id = 103, sorted by last name:

<u>aa.go.</u>	<u> 100, 001</u>	tou by last_nai		
A	EMPLOYEE_ID	LAST_NAME	SALARY 2	DEPARTMENT_ID
1	104	Ernst	6000	60
2	107	Lorentz	4200	60

manager id = 201, sorted by salary:

		a		\ \
2	EMPLOYEE_ID	LAST_NAME	E SALARY	DEPARTMENT_ID
1	202	Fay	6000	20

manager_id = 124, sorted by employee_id:

	A	EMPLOYEE_ID	LAST_NAME	A	SALARY	DEPARTMENT_ID
1		141	Rajs		3500	50
2		142	Davies		3100	50
3		143	Matos		2600	50
4		144	Vargas		2500	50