

# SQL Exercises Chapter 5,6,7

Anik Kirtania

Date: 2025-05-31

## Chapter 5

1. **Question:**

Group functions work across many rows to produce one result per group. True or False?

**Answer:** True

2. **Question:**

Group functions include nulls in calculations. True or False?

**Answer:** False

*Explanation: Group functions automatically exclude NULL values from calculations unless specifically included using NVL or COALESCE functions.*

3. **Question:**

The WHERE clause restricts rows before inclusion in a group calculation. True or False?

**Answer:** True

*Explanation: WHERE clause filters rows before they are grouped, while HAVING filters after grouping.*

4. **Question:**

The HR department needs the following reports: Find the highest, lowest, sum, and average salary of all employees. Label the columns as Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number.

**Command:**

```
1      SELECT MAX(salary) AS Maximum,
2             MIN(salary) AS Minimum,
3             SUM(salary) AS Sum,
4             ROUND(AVG(salary)) AS Average
5      FROM employees;
```

5. **Question:**

Modify the query to display the minimum, maximum, sum, and average salary for each job type.

**Command:**

```
1      SELECT job_id,
2             MAX(salary) AS Maximum,
3             MIN(salary) AS Minimum,
4             SUM(salary) AS Sum,
5             ROUND(AVG(salary)) AS Average
6      FROM employees
7      GROUP BY job_id
8      ORDER BY job_id;
```

6. **Question:**

Write a query to display the number of people with the same job.

**Command:**

```
1      SELECT job_id,
```

```
2          COUNT(*) AS COUNT
3  FROM employees
4  GROUP BY job_id
5  ORDER BY job_id;
```

**Generalized Query:**

```
1  SELECT job_id,
2          COUNT(*) AS COUNT
3  FROM employees
4  WHERE job_id = '&job_title'
5  GROUP BY job_id;
```

**7. Question:**

Determine the number of managers without listing them. Label the column as Number of Managers. Hint: Use the MANAGER\_ID column to determine the number of managers.

**Command:**

```
1  SELECT COUNT(DISTINCT manager_id) AS "Number of Managers"
2  FROM employees;
```

**8. Question:**

Find the difference between the highest and lowest salaries. Label the column DIFFERENCE.

**Command:**

```
1  SELECT MAX(salary) - MIN(salary) AS DIFFERENCE
2  FROM employees;
```

**9. Question:**

Create a report to display the manager number and the salary of the lowest-paid employee for that manager. Exclude anyone whose manager is not known. Exclude any groups where the minimum salary is 6,000 or less. Sort the output in descending order of salary.

**Command:**

```
1  SELECT manager_id, MIN(salary) AS MIN_SALARY
2  FROM employees
3  WHERE manager_id IS NOT NULL
4  GROUP BY manager_id
5  HAVING MIN(salary) > 6000
6  ORDER BY MIN(salary) DESC;
```

**10. Question:**

Create a query to display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998. Create appropriate headings.

**Command:**

```
1  SELECT COUNT(*) AS TOTAL,
2      SUM(CASE WHEN TO_CHAR(hire_date, 'YYYY') = '1995' THEN 1 ELSE
3          0 END) AS "1995",
4      SUM(CASE WHEN TO_CHAR(hire_date, 'YYYY') = '1996' THEN 1 ELSE
5          0 END) AS "1996",
6      SUM(CASE WHEN TO_CHAR(hire_date, 'YYYY') = '1997' THEN 1 ELSE
7          0 END) AS "1997",
8      SUM(CASE WHEN TO_CHAR(hire_date, 'YYYY') = '1998' THEN 1 ELSE
9          0 END) AS "1998"
10 FROM employees;
```

### 11. Question:

Create a matrix query to display the job, the salary for that job based on department number, and the total salary for that job, for departments 20, 50, 80, and 90, giving each column an appropriate heading.

**Command:**

```
1  SELECT job_id,
2      SUM(DECODE(department_id, 20, salary)) AS "Dept 20",
3      SUM(DECODE(department_id, 50, salary)) AS "Dept 50",
4      SUM(DECODE(department_id, 80, salary)) AS "Dept 80",
5      SUM(DECODE(department_id, 90, salary)) AS "Dept 90",
6      SUM(salary) AS "Total"
7  FROM employees
8  GROUP BY job_id
9  ORDER BY job_id;
```

## Chapter 6

### 1. Question:

Write a query for the HR department to produce the addresses of all the departments. Use the LOCATIONS and COUNTRIES tables. Show the location ID, street address, city, state or province, and country in the output. Use a NATURAL JOIN to produce the results.

**Command:**

```
1  SELECT l.location_id,
2      l.street_address,
3      l.city,
4      l.state_province,
5      c.country_name
6  FROM locations l
7  NATURAL JOIN countries c
8  ORDER BY location_id;
```

### 2. Question:

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

**Command:**

```
1  SELECT e.last_name ,
2         d.department_id ,
3         d.department_name
4  FROM employees e
5  JOIN departments d ON e.department_id = d.department_id
6  ORDER BY e.employee_id;
```

**3. Question:**

The HR department needs a report of employees in Toronto. Display the last name, job, department number, and the department name for all employees who work in Toronto.

**Command:**

```
1  SELECT e.last_name ,
2         e.job_id ,
3         d.department_id ,
4         d.department_name
5  FROM employees e
6  JOIN departments d ON e.department_id = d.department_id
7  JOIN locations l ON d.location_id = l.location_id
8  WHERE l.city = 'Toronto';
```

**4. Question:**

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. Save your SQL statement as lab\_06\_04.sql.

**Command:**

```
1  SELECT e.last_name AS "Employee",
2         e.employee_id AS "Emp#",
3         m.last_name AS "Manager",
4         e.manager_id AS "Mgr#"
5  FROM employees e
6  LEFT JOIN employees m ON e.manager_id = m.employee_id
7  ORDER BY e.employee_id;
```

**5. Question:**

Modify lab\_06\_04.sql to display all employees including King, who has no manager. Order the results by employee number. Save your SQL statement as lab\_06\_05.sql.

**Command:**

```
1  SELECT e.last_name AS "Employee",
2         e.employee_id AS "Emp#",
3         NVL(m.last_name, '(null)') AS "Manager",
4         e.manager_id AS "Mgr#"
5  FROM employees e
6  LEFT OUTER JOIN employees m ON e.manager_id = m.employee_id
7  ORDER BY e.employee_id;
```

**6. Question:**

Create a report for the HR department that displays employee last names, department numbers, and all the employees who work in the same department as a given employee. Give each column an appropriate label. Save the script to a file named lab\_06\_06.sql.

**Command:**

```
1      SELECT DISTINCT d.department_id AS "DEPARTMENT",
2                      e1.last_name AS "EMPLOYEE",
3                      e2.last_name AS "COLLEAGUE"
4      FROM employees e1
5      JOIN employees e2 ON e1.department_id = e2.department_id
6      JOIN departments d ON e1.department_id = d.department_id
7      WHERE e1.employee_id != e2.employee_id
8      ORDER BY d.department_id, e1.last_name, e2.last_name;
```

**7. Question:**

The HR department needs a report on job grades and salaries. To familiarize yourself with the JOB\_GRADES table, first show the structure of the JOB\_GRADES table. Then create a query that displays the name, job, department name, salary, and grade for all employees.

**Query Command:**

```
1      SELECT e.last_name ,
2             e.job_id ,
3             d.department_name ,
4             e.salary ,
5             j.grade_level
6      FROM employees e
7      JOIN departments d ON e.department_id = d.department_id
8      JOIN job_grades j ON e.salary
9                      BETWEEN j.lowest_sal AND j.highest_sal
10     ORDER BY e.salary;
```

**8. Question:**

The HR department wants to determine the names of all the employees who were hired after Davies. Create a query to display the name and hire date of any employee hired after employee Davies.

**Command:**

```
1      SELECT last_name , hire_date
2      FROM employees
3      WHERE hire_date > (
4          SELECT hire_date
5          FROM employees
6          WHERE last_name = 'Davies'
7      )
8      ORDER BY hire_date;
```

**9. Question:**

The HR department needs to find the names and hire dates of all the employees who were hired before their managers, along with their managers' names and hire dates. Save the script to a file named lab\_06\_09.sql.

**Command:**

```
1  SELECT
2      e.last_name ,
3      e.hire_date ,
4      m.last_name AS "MANAGER_NAME",
5      m.hire_date AS "MGR_HIRE_DATE"
6  FROM employees e
7  JOIN employees m ON e.manager_id = m.employee_id
8  WHERE e.hire_date < m.hire_date
9  ORDER BY e.hire_date;
```

## Chapter 7

### 1. Question:

The HR department needs a query that prompts the user for an employee last name. The query then displays the last name and hire date of any employee in the same department as the employee whose name they supply (excluding that employee). For example, if the user enters Zlotkey, find all employees who work with Zlotkey (excluding Zlotkey).

**Command:**

```
1  SELECT last_name , hire_date
2  FROM employees
3  WHERE department_id = (
4      SELECT department_id
5      FROM employees
6      WHERE last_name = '&ENTER_NAME'
7  )
8  AND last_name != '&ENTER_NAME';
```

**Prompt:**

ENTER\_NAME: Zlotkey

### 2. Question:

Create a report that displays the employee number, last name, and salary of all employees who earn more than the average salary. Sort the results in order of ascending salary.

**Command:**

```
1  SELECT employee_id ,
2      last_name ,
3      salary
4  FROM employees
5  WHERE salary > (
6      SELECT AVG(salary)
7      FROM employees
8  )
9  ORDER BY salary ASC;
```

**3. Question:**

Write a query that displays the employee number and last name of all employees who work in a department with any employee whose last name contains the letter "u". Save your SQL statement as lab\_07.03.sql.

**Command:**

```
1  SELECT DISTINCT e.employee_id, e.last_name
2  FROM employees e
3  JOIN employees u
4  ON e.department_id = u.department_id
5  WHERE u.last_name LIKE '%u%'
6  ORDER BY e.employee_id;
```

**4. Question:**

The HR department needs a report that displays the last name, department number, and job ID of all employees whose department location ID is 1700.

**Command:**

```
1  SELECT e.last_name, e.department_id, e.job_id
2  FROM employees e
3  JOIN departments d ON e.department_id = d.department_id
4  WHERE d.location_id = 1700
5  ORDER BY e.last_name;
```

**5. Question:**

Create a report for HR that displays the last name and salary of every employee who reports to King.

**Command:**

```
1  SELECT last_name, salary
2  FROM employees
3  WHERE manager_id = (
4      SELECT employee_id
5      FROM employees
6      WHERE last_name = 'King'
7  )
8  ORDER BY salary DESC;
```

**6. Question:**

Create a report for HR that displays the department number, last name, and job ID for every employee in the Executive department.

**Command:**

```
1  SELECT e.department_id,
2         e.last_name,
3         e.job_id
4  FROM employees e
5  JOIN departments d ON e.department_id = d.department_id
6  WHERE d.department_name = 'Executive'
```



```
7      ORDER BY e.department_id;
```

**7. Question:**

Modify the query in lab\_07\_03.sql to 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". Save as lab\_07\_07.sql.

**Command:**

```
1      SELECT DISTINCT e.employee_id, e.last_name, e.salary
2      FROM employees e
3      JOIN employees u ON e.department_id = u.department_id
4      WHERE u.last_name LIKE '%u%'
5      AND e.salary > (SELECT AVG(salary) FROM employees)
6      ORDER BY e.employee_id;
```