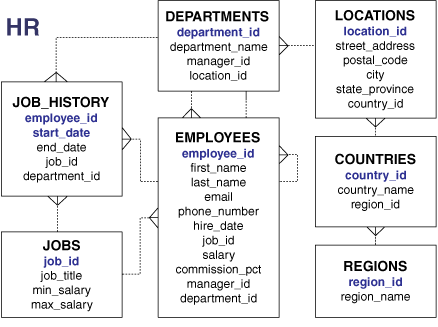
## **Objectives:**

The aim of this lab is to exercise the group function that exist in SQL. Also, we introduce the group by clause and having clause.

## **Lab Work:**

The following SQL queries are based on the HR database with the schema shown below.



The general form of a select statement:

SELECT *column*, *group\_function*

FROM *table*

[WHERE *condition*]

[GROUP BY *group\_by\_expression*]

[HAVING *group\_condition*]

[ORDER BY *column*];

The Oracle server evaluates the clauses in the following order:

1. If the statement contains a WHERE clause, the server establishes the candidate rows.

2. The server identifies the groups that are specified in the GROUP BY clause.

3. The HAVING clause further restricts result groups that do not meet the group criteria in

the HAVING clause.

1. Display the average, highest, lowest, and sum of monthly salaries for all sales representatives.

SELECT AVG(salary), MAX(salary),

MIN(salary), SUM(salary)

FROM employees

WHERE job\_id LIKE '%REP%';

1. Display the employee last name that is first and the employee last name that is last in an alphabetized list of all employees:

SELECT MIN(last\_name), MAX(last\_name)

FROM employees;

1. Displays the number of employees in department 50

SELECT COUNT(\*)

FROM employees

WHERE department\_id = 50;

1. displays the number of employees in department 80 who can earn a commission.

SELECT COUNT(commission\_pct)

FROM employees

WHERE department\_id = 80;

1. Display the number of distinct department values that are in the EMPLOYEES table.

SELECT COUNT (DISTINCT department\_id)

FROM employees;

1. Calculate the average commision based on all rows in the table, regardless of whether null values are stored in the COMMISSION\_PCT column

SELECT AVG (NVL (commission\_pct, 0))

FROM employees;

1. Display the department number and the average salary for each department.

SELECT department\_id, AVG(salary)

FROM employees

GROUP BY department\_id;

SELECT department\_id, AVG(salary)

FROM employees

GROUP BY department\_id

ORDER BY AVG(salary);

1. Display the department\_id, job\_id and sum of all salaries within grouped by department and job

SELECT department\_id dept\_id, job\_id, SUM(salary)

FROM employees

GROUP BY department\_id, job\_id;

1. Display of average salaries of those departments that have an average salary greater than $8,000.

SELECT department\_id, AVG(salary)

FROM employees

HAVING AVG(salary) > 8000

GROUP BY department\_id;

1. Display the job ID and total monthly salary for each job that has a total payroll exceeding $13,000. Excludes sales representatives

SELECT job\_id, SUM(salary) PAYROLL

FROM employees

WHERE job\_id NOT LIKE '%REP%'

GROUP BY job\_id

HAVING SUM(salary) > 13000;

1. Display the maximum average salary:

SELECT MAX(AVG(salary))

FROM employees

GROUP BY department\_id;

**Notes:**

- Any column or expression in the SELECT list that is not

an aggregate function must be in the GROUP BY clause.

- You cannot use the WHERE clause to restrict groups.

- You use the HAVING clause to restrict groups.

- You cannot use group functions in the WHERE clause.

## **Class Exercise:**

1. Find the highest, lowest, sum, and average salary of all employees. Label the columnsMaximum, Minimum, Sum, and Average, respectively. Round your results to the Nearest whole number.
2. Modify the query in 1 to display the minimum, maximum, sum, andaverage salary for each job type.
3. Write a query to display the number of people with the same job.
4. Determine the number of managers without listing them. Label the column Numberof Managers. Hint: Use the MANAGER\_ID column to determine the number ofmanagers.
5. Find the difference between the highest and lowest salaries. Label the columnDIFFERENCE.
6. Create a report to display the manager number and the salary of the lowest-paidemployee for that manager. Exclude anyone whose manager is not known. Exclude anygroups where the minimum salary is $6,000 or less. Sort the output in descending orderof salary.
7. Create a query to display the total number of employees and, of that total, the number ofemployees hired in 1995, 1996, 1997, and 1998.
8. Create a matrix query to display the job, the salary for that job based on departmentnumber, and the total salary for that job, for departments 20, 50, 80, and 90, giving each column an appropriate heading.