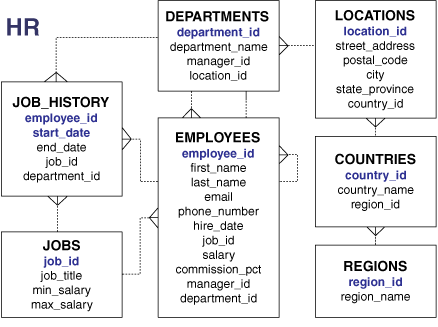
## **Objectives:**

The aim of this lab is to practice extracting data from more than one table. We do this based on the HR database described below. This include Equijoins, Non-equijoins, Outer joins, Self-joins

Cross joins, Natural joins, Full (or two-sided) outer joins.

## **Lab Work:**



1. Retrieve Department name, their cities and location IDs for department whose number is either 20 or 50:

SELECT department\_id, department\_name,

location\_id, city

FROM departments

NATURAL JOIN locations

WHERE department\_id IN (20, 50);

Alternative notation would be

SELECT l.city, d.department\_name

FROM locations l JOIN departments d USING (location\_id)

Where department\_id in (20,50);

1. Display employees last name, id, department location and department id to which he belongs

SELECT employees.employee\_id, employees.last\_name,

departments.location\_id, department\_id

FROM employees JOIN departments

USING (department\_id);

Alternative

SELECT employees.employee\_id, employees.last\_name,

departments.department\_id, departments.location\_id

FROM employees JOIN departments

ON employees.department\_id = departments.department\_id;

To simplify, we can use aliases as follows:

SELECT e.employee\_id, e.last\_name,

d.location\_id, department\_id

FROM employees e JOIN departments d

USING (department\_id) ;

An alternative using ON clause

SELECT e.employee\_id, e.last\_name, e.department\_id,

d.department\_id, d.location\_id

FROM employees e JOIN departments d

ON (e.department\_id = d.department\_id);

1. Display the names of the employees and their managers.

SELECT e.last\_name emp, m.last\_name mgr

FROM employees e JOIN employees m

ON (e.manager\_id = m.employee\_id);

1. Display info from employees and department table for the manager whose id =149:

SELECT e.employee\_id, e.last\_name, e.department\_id,

d.department\_id, d.location\_id

FROM employees e JOIN departments d

ON (e.department\_id = d.department\_id)

AND e.manager\_id = 149 ;

1. Join between three tables:

SELECT employee\_id, city, department\_name

FROM employees e

JOIN departments d

ON d.department\_id = e.department\_id

JOIN locations l

ON d.location\_id = l.location\_id;

1. Create a non-equijoin to evaluate an employee’s salary grade. The salary

must be *between* any pair of the low and high salary ranges.

SELECT e.last\_name, e.salary,j.min\_salary, j.max\_salary

FROM employees e JOIN jobs j

ON e.salary

BETWEEN j.min\_salary AND j.max\_salary;

1. Retrieves all rows in the EMPLOYEES table, which is the left table even if there

is no match in the DEPARTMENTS table.

SELECT e.last\_name, e.department\_id, d.department\_name

FROM employees e LEFT OUTER JOIN departments d

ON (e.department\_id = d.department\_id) ;

1. Retrieves all rows in the DEPARTMENTS table, which is the right table even if there is no match in the EMPLOYEES table.

SELECT e.last\_name, e.department\_id, d.department\_name

FROM employees e RIGHT OUTER JOIN departments d

ON (e.department\_id = d.department\_id) ;

1. Retrieves all rows in the EMPLOYEES table, even if there is no match in the

DEPARTMENTS table. Also retrieves all rows in the DEPARTMENTS table, even if

there is no match in the EMPLOYEES table.

SELECT e.last\_name, d.department\_id, d.department\_name

FROM employees e FULL OUTER JOIN departments d

ON (e.department\_id = d.department\_id) ;

1. Produces a Cartesian product of the EMPLOYEES and

DEPARTMENTS tables.

SELECT last\_name, department\_name

FROM employees

CROSS JOIN departments;

## **Class Exercise:**

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

2. 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.

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

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

5. Display all employees including King, who has no manager.

Order the results by the employee number.

6. Create a report for the HR department that displays employee last names, departmentnumbers, and all the employees who work in the same department as a given employee. Giveeach column an appropriate label.

7. The HR department needs a report on job grades and salaries.

Then create a query that displays the name, job, department name, salary, and grade for all employees.

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

9. 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.