Joins

Learning Objective

After completing this lab the student should be able to:

- What is joins why we use it ?
- Difference between inner join and outer join.
- Outer join types (Left join, Right join, full join).
- Self join and cross join.
- Practical example and exercises.

Tools and Technologies

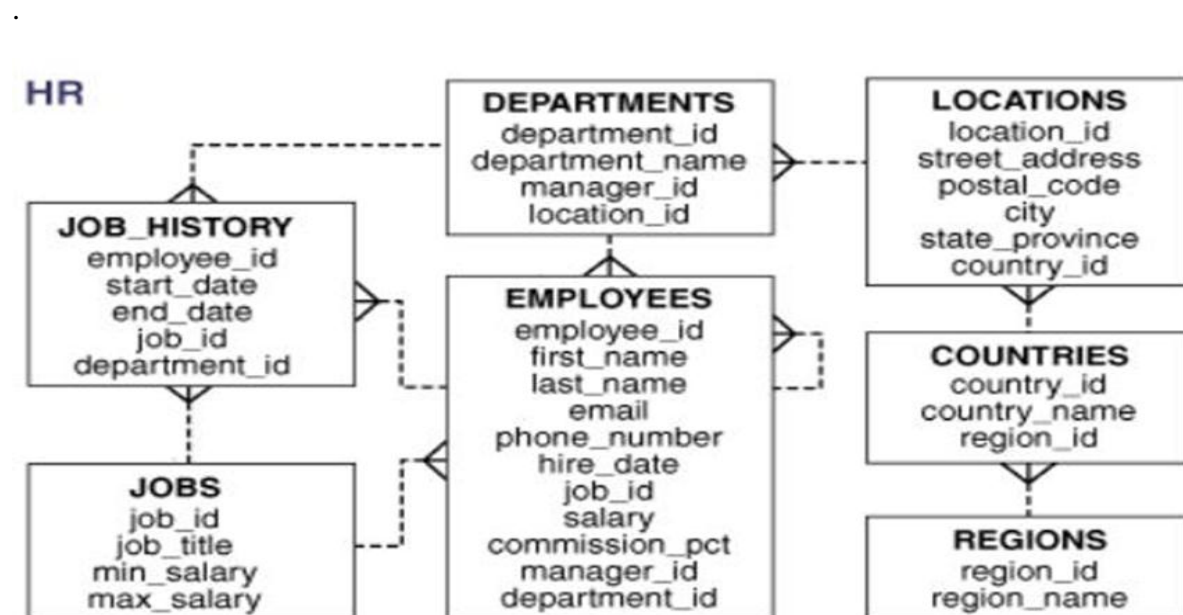- Oracle Database 11g Express Edition/Enterprise Edition.

Oracle Credentials for Lab

Enter the Url in your browser **http:\\172.168.8.16:8080\apex**

Username **hr**

Password **hr**

HR Schema

.

HR Table Descriptions

```
Table COUNTRIES
 Name                                      Null?    Type
 ----------------------------------------- -------- ---------------
 COUNTRY_ID                                NOT NULL CHAR(2)
 COUNTRY_NAME                                       VARCHAR2(40)
 REGION_ID                                          NUMBER

Table DEPARTMENTS
 Name                                      Null?    Type
 ----------------------------------------- -------- ---------------
 DEPARTMENT_ID                             NOT NULL NUMBER(4)
 DEPARTMENT_NAME                           NOT NULL VARCHAR2(30)
 MANAGER ID                                         NUMBER(6)
 LOCATION ID                                        NUMBER(4)

Table EMPLOYEES
 Name                                      Null?    Type
 ----------------------------------------- -------- ---------------
 EMPLOYEE_ID                               NOT NULL NUMBER(6)
 FIRST_NAME                                         VARCHAR2(20)
 LAST_NAME                                 NOT NULL VARCHAR2(25)
 EMAIL                                     NOT NULL VARCHAR2(25)
 PHONE_NUMBER                                       VARCHAR2(20)
 HIRE_DATE                                          NOT NULL DATE
 JOB_ID                                             NOT NULL
VARCHAR2(10)
 SALARY                                             NUMBER(8,2)
 COMMISSION PCT                                     NUMBER(2,2)
 MANAGER ID                                         NUMBER(6)
 DEPARTMENT_ID                                      NUMBER(4)

Table JOBS
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 JOB ID                                    NOT NULL VARCHAR2(10)
 JOB_TITLE                                 NOT NULL VARCHAR2(35)
 MIN_SALARY                                         NUMBER(6)
 MAX_SALARY                                         NUMBER(6)

Table JOB_HISTORY
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 EMPLOYEE ID                               NOT NULL NUMBER(6)
 START DATE                                NOT NULL DATE
 END_DATE                                  NOT NULL DATE
 JOB_ID                                    NOT NULL VARCHAR2(10)
 DEPARTMENT ID                                      NUMBER(4)

Table LOCATIONS
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------
 LOCATION ID                               NOT NULL NUMBER(4)
 STREET_ADDRESS                                     VARCHAR2(40)
 POSTAL_CODE                                        VARCHAR2(12)
 CITY                                      NOT NULL VARCHAR2(30)
 STATE_PROVINCE                                     VARCHAR2(25)
 COUNTRY_ID                                         CHAR(2)
```

```
Table REGIONS
 Name                                      Null?     Type
 ----------------------------------------- --------  ---------------
 REGION ID                                 NOT NULL  NUMBER
 REGION NAME                                         VARCHAR2(25)
```

SQL: Joins

Introduction

A join is a query that combines rows from two or more tables, views, or materialized views. Oracle Database performs a join whenever multiple tables appear in the FROM clause of the query. The select list of the query can select any columns from any of these tables. If any two of these tables have a column name in common, then you must qualify all references to these columns throughout the query with table names to avoid ambiguity.

The different types of SQL joins are:

- INNER JOIN (or sometimes called simple join)
- LEFT OUTER JOIN (or sometimes called LEFT JOIN)
- RIGHT OUTER JOIN (or sometimes called RIGHT JOIN)
- FULL OUTER JOIN (or sometimes called FULL JOIN)
- SELF JOIN
- CROSS JOIN

INNER JOIN (simple join)

An inner join (sometimes called a simple join) is a join of two or more tables that returns only those rows that satisfy the join condition.
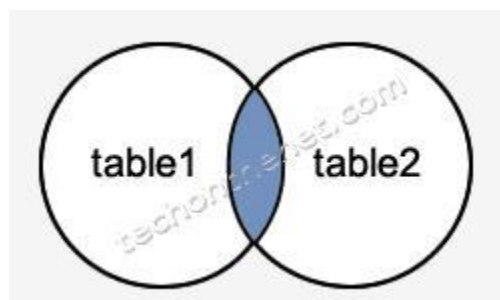
Syntax

SELECT columns FROM table1

INNER JOIN table2 ON table1.column = table2.column;

Illustration of an INNER JOIN

An INNER JOIN returns the shaded area:

Example

```
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);
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID	DEPARTMENT_ID	LOCATION_ID
200	Whalen	10	10	1700
201	Hartstein	20	20	1800
202	Fay	20	20	1800
124	Mourgos	50	50	1500
141	Rajs	50	50	1500
142	Davies	50	50	1500
143	Matos	50	50	1500

This SQL INNER JOIN example would return all rows from the employees and departments tables where there is a matching ,department_id value in both the employees and departments tables.

LEFT OUTER JOIN or LEFT JOIN

The LEFT JOIN keyword returns all records from the left table (table1), and the matched records from the right table (table2). The result is NULL from the right side, if there is no match.

Illustration of a LEFT OUTER JOIN

Example

```
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);
```

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Fay	20	Marketing
Hartstein	20	Marketing

| De Haan	90	Executive
Kochhar	90	Executive
King	90	Executive
Gietz	110	Accounting
Higgins	110	Accounting
Grant		

20 rows selected.

Right Outer join

The RIGHT JOIN keyword returns all records from the right table (table2), and the matched records from the left table (table1). The result is NULL from the left side, when there is no match**.**

Syntax

```
SELECT columns
FROM table1
RIGHT [OUTER] JOIN table2
ON table1.column = table2.column;
```

Illustration of a RIGHT OUTER JOIN

A RIGHT OUTER JOIN returns the shaded area:
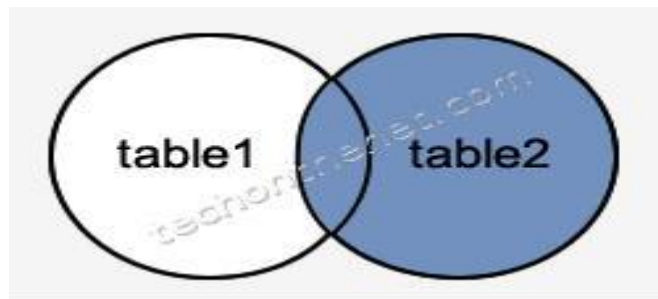
Example

```
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);
```

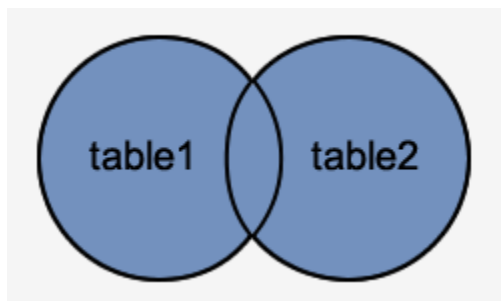| LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Fay	20	Marketing
Hartstein	20	Marketing
Davies	50	Shipping
Kochhar	90	Executive
Gietz	110	Accounting
Higgins	110	Accounting

Full Outer join

The FULL OUTER JOIN keyword return all records when there is a match in either left (table1) or right (table2) table records.

Syntax

```
SELECT column_names
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name;
```

Illustration of a Full OUTER JOIN

Example

```
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);
```

Output

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
Whalen	10	Administration
Fay	20	Marketing
Hartstein	20	Marketing

LAST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
King	90	Executive
Gietz	110	Accounting
Higgins	110	Accounting
Grant		
	190	Contracting

21 rows selected.

Self Join

A self-join is a join in which a table is joined with itself (which is also called Unary relationships), especially when the table has a FOREIGN KEY which references its own PRIMARY KEY. To join a table itself means that each row of the table is combined with itself and with every other row of the table.
The self-join can be viewed as a join of two copies of the same table. The table is not actually copied, but SQL performs the command as though it were.

Syntax

SELECT a.column_name, b.column_name
FROM table1 a, table1 b
ON a.common_filed = b.common_field;

Example

SELECT e.last_name emp, m.last_name mgr
FROM employees e JOIN employees m
ON (e.manager_id = m.employee_id);

EMPLOYEE_ID	LAST_NAME	MANAGER_ID
100	King	
101	Kochhar	100
102	De Haan	100
103	Hunold	102
104	Ernst	103
107	Lorentz	103
124	Mourgos	100

EMPLOYEE_ID	LAST_NAME
100	King
101	Kochhar
102	De Haan
103	Hunold
104	Ernst
107	Lorentz
124	Mourgos

MANAGER_ID in the WORKER table is equal to
EMPLOYEE_ID in the MANAGER table.

Applying Additional Conditions to a Join

Example

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 ;

Creating Three-Way Joins with the ON Clause

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;

EMPLOYEE_ID	CITY	DEPARTMENT_NAME
103	Southlake	IT
104	Southlake	IT
107	Southlake	IT
124	South San Francisco	Shipping
141	South San Francisco	Shipping
142	South San Francisco	Shipping
143	South San Francisco	Shipping
144	South San Francisco	Shipping

Cross join

CROSS JOIN returns the Cartesian product of rows from tables in the join. In other words,
it will produce rows which combine each row from the first table with each row from the
second table.

Example

SELECT last_name, department_name
FROM employees
CROSS JOIN departments;

Lab Exercise

Q1 Implement your Lab example here.

Q2 Write a query that displays the job-title, employee last name, department id for all employees whose employee job-id is equal to job job-id.

Q3 Write a query 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.

Lab Instructor:

Qazi Shuja Ud Din (Teaching Fellow) (Riphah International University)
Email: qazi.shujauddin@riphah.edu.pk