



**Make A Free Website**  
Free Hosting, Design Tools and More...

**Click Here**

WebStarts.com

## 7.4.1. Understanding Outer Joins

1. An outer join retrieves a row even when one of the columns in the join contains a null value.
2. You perform an outer join by supplying the outer join operator in the join condition.
3. The outer join operator is a plus character in parentheses (+).
4. The outer join operator (+) is on the column that contains the null value.

```
SQL> -- create demo table
SQL> create table Employee(
2     EMPNO      NUMBER(3),
3     ENAME      VARCHAR2(15 BYTE),
4     HIREDATE   DATE,
5     ORIG_SALARY NUMBER(6),
6     CURR_SALARY NUMBER(6),
7     REGION     VARCHAR2(1 BYTE)
8 )
9 /

Table created.

SQL>
SQL> create table job (
2     EMPNO      NUMBER(3),
3     jobtitle   VARCHAR2(20 BYTE)
4 )
5 /

Table created.

SQL>
SQL> insert into job (EMPNO, Jobtitle) values (1,'Tester');

1 row created.

SQL> insert into job (EMPNO, Jobtitle) values (2,'Accountant');

1 row created.

SQL> insert into job (EMPNO, Jobtitle) values (3,'Developer');

1 row created.

SQL> insert into job (EMPNO, Jobtitle) values (4,'CODer');

1 row created.

SQL> insert into job (EMPNO, Jobtitle) values (5,'Director');

1 row created.

SQL>
SQL> insert into job (EMPNO, Jobtitle) values (9,'Developer');

1 row created.

SQL>
SQL> -- prepare data
SQL> insert into Employee(EMPNO, EName, HIREDATE, ORIG_SALARY, CURR_SALARY, REGION)
2     values (1, 'Jason', to_date('19960725','YYYYMMDD'), 1234, 8767, 'E')
3 /

1 row created.

SQL> insert into Employee(EMPNO, EName, HIREDATE, ORIG_SALARY, CURR_SALARY, REGION)
2     values (2, 'John', to_date('19970715','YYYYMMDD'), 2341, 3456, 'W')
3 /

1 row created.

SQL> insert into Employee(EMPNO, EName, HIREDATE, ORIG_SALARY, CURR_SALARY, REGION)
2     values (3, 'Joe', to_date('19860125','YYYYMMDD'), 4321, 5654, 'E')
3 /

1 row created.

SQL> insert into Employee(EMPNO, EName, HIREDATE, ORIG_SALARY, CURR_SALARY, REGION)
2     values (4, 'Tom', to_date('20060913','YYYYMMDD'), 2413, 6787, 'W')
3 /

1 row created.

SQL> insert into Employee(EMPNO, EName, HIREDATE, ORIG_SALARY, CURR_SALARY, REGION)
2     values (5, 'Jane', to_date('20050417','YYYYMMDD'), 7654, 4345, 'E')
```

```
3 /
1 row created.

SQL> insert into Employee(EMPNO, EName, HIREDATE, ORIG_SALARY, CURR_SALARY, REGION)
2 values (6, 'James', to_date('20040718','YYYYMMDD'), 5679, 6546, 'W')
3 /

1 row created.

SQL> insert into Employee(EMPNO, EName, HIREDATE, ORIG_SALARY, CURR_SALARY, REGION)
2 values (7, 'Jodd', to_date('20030720','YYYYMMDD'), 5438, 7658, 'E')
3 /

1 row created.

SQL> insert into Employee(EMPNO, EName, HIREDATE, ORIG_SALARY, CURR_SALARY, REGION)
2 values (8, 'Joke', to_date('20020101','YYYYMMDD'), 8765, 4543, 'W')
3 /

1 row created.

SQL> insert into Employee(EMPNO, EName, HIREDATE, ORIG_SALARY, CURR_SALARY, REGION)
2 values (9, 'Jack', to_date('20010829','YYYYMMDD'), 7896, 1232, 'E')
3 /

1 row created.

SQL>
SQL>
SQL>
SQL> -- display data in the table
SQL> select * from Employee
2 /

EMPNO ENAME HIREDATE ORIG_SALARY CURR_SALARY R
-----
1 Jason 25-JUL-96 1234 8767 E
2 John 15-JUL-97 2341 3456 W
3 Joe 25-JAN-86 4321 5654 E
4 Tom 13-SEP-06 2413 6787 W
5 Jane 17-APR-05 7654 4345 E
6 James 18-JUL-04 5679 6546 W
7 Jodd 20-JUL-03 5438 7658 E
8 Joke 01-JAN-02 8765 4543 W
9 Jack 29-AUG-01 7896 1232 E

9 rows selected.

SQL> select * from job
2 /

EMPNO JOBTITLE
-----
1 Tester
2 Accountant
3 Developer
4 Coder
5 Director
9 Developer

6 rows selected.

SQL>
SQL>
SQL> SELECT e.ename, j.jobtitle FROM employee e, job j WHERE e.empno = j.empno (+);

ENAME JOBTITLE
-----
Jason Tester
John Accountant
Joe Developer
Tom Coder
Jane Director
Jack Developer
Joke
James
Jodd

9 rows selected.

SQL>
SQL> SELECT e.ename, j.jobtitle FROM employee e, job j WHERE e.empno (+) = j.empno;

ENAME JOBTITLE
-----
Jason Tester
John Accountant
Joe Developer
Tom Coder
Jane Director
Jack Developer

6 rows selected.

SQL>
SQL> SELECT e.ename, j.jobtitle FROM employee e, job j WHERE j.empno (+) = e.empno;
```

```

ENAME          JOBTITLE
-----
Jason          Tester
John           Accountant
Joe            Developer
Tom            COder
Jane           Director
Jack           Developer
Joke
James
Jodd

9 rows selected.

SQL>
SQL> -- clean the table
SQL> drop table Employee
2 /

Table dropped.

SQL> drop table job
2 /

Table dropped.

SQL>

```



#### [Ceramic Paper](http://www.VitCas.com/Ceramic+Paper)

[www.VitCas.com/Ceramic+Paper](http://www.VitCas.com/Ceramic+Paper)  
UK Manufacturer of Ceramic Fibre  
Adhesive, High Quality Refractories



#### [Learn More on SQL Server](http://www.sswug.org)

[www.sswug.org](http://www.sswug.org)  
Access Videos, Articles and More.  
SSWUG Community & vConference



#### [Pl Sql Expert](http://www.allroundautomations.com)

[www.allroundautomations.com](http://www.allroundautomations.com)  
Download the 30 day trail version for  
PL/SQL IDE!



AdChoices

### 7.4.Outer Joins Left Right

- 7.4.1. [Understanding Outer Joins](#)
- 7.4.2. [Left and Right Outer Joins](#)
- 7.4.3. [An Example of a Left Outer Join 1](#)
- 7.4.4. [An Example of a Left Outer Join 2](#)
- 7.4.5. [An Example of a Right Outer Join 1](#)
- 7.4.6. [An Example of a Right Outer Join 2](#)
- 7.4.7. [Perform outer joins in combination with self joins, employee and job tables](#)
- 7.4.8. [Example outer join with \(+\)](#)
- 7.4.9. [Right outer join with using statement](#)
- 7.4.10. [Right outer join with group by](#)
- 7.4.11. [LEFT OUTER JOIN tableName ON joined columns](#)
- 7.4.12. [LEFT OUTER JOIN vs RIGHT OUTER JOIN](#)
- 7.4.13. [Left Outer Join](#)
- 7.4.14. [Right Outer Join](#)
- 7.4.15. [Right Outer Join\(room vs class\)](#)
- 7.4.16. [Right join with where in clause](#)

[java2s.com](http://java2s.com) | [Contact Us](#) | [Privacy Policy](#)

Copyright 2009 - 12 Demo Source and Support. All rights reserved.  
All other trademarks are property of their respective owners.