

1. Introduction
2. Query Select
3. Set
4. Insert Update
Delete
5. Sequences
6. Table
7. **Table Joins**
8. View
9. Index
10. SQL Data Types
11. Character String
Functions
12. Aggregate
Functions
13. Date Timestamp
Functions
14. Numerical Math
Functions
15. Conversion
Functions
16. Analytical Functions
17. Miscellaneous
Functions
18. Regular
Expressions
Functions
19. Statistical
Functions
20. Linear Regression
Functions
21. PL SQL Data
Types
22. PL SQL
Statements
23. PL SQL Operators
24. PL SQL
Programming
25. Cursor
26. Collections
27. Function Procedure
Packages
28. Trigger
29. SQL PLUS
Session
Environment
30. System Tables
Data Dictionary
31. System Packages
32. Object Oriented
33. XML
34. Large Objects
35. Transaction
36. User Privilege



7.5.2. You cannot use an outer join condition with the IN operator

```
SQL>
SQL>
SQL>
SQL>
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> insert into job (EMPNO, Jobtitle) values (6,'Mediator');

1 row created.

SQL> insert into job (EMPNO, Jobtitle) values (7,'Proffessor');

1 row created.

SQL> insert into job (EMPNO, Jobtitle) values (8,'Programmer');

1 row created.

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

1 row created.

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

1 row created.

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

1 row created.

SQL> insert into Employee(EMPNO, EName, HIREDATE, ORIG_SALARY, CURF
```



```

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

1 row created.

SQL> insert into Employee(EMPNO, EName, HIREDATE, ORIG_SALARY, CURRF
2      values (5, 'Jane', to_date('20050417','YYYYMMDD'), 7654,
3      /

1 row created.

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

1 row created.

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

1 row created.

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

1 row created.

SQL> insert into Employee(EMPNO, EName, HIREDATE, ORIG_SALARY, CURRF
2      values (9, 'Jack', to_date('20010829','YYYYMMDD'), 7896,
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
6 Mediator
7 Proffessor
8 Programmer
9 Developer

9 rows selected.

SQL>
SQL> SELECT e.ename, j.jobtitle
2      FROM employee e, job j
3      WHERE e.empno (+) IN (1, 2, 3, 4);
      WHERE e.empno (+) IN (1, 2, 3, 4)
      *

ERROR at line 3:
ORA-01719: outer join operator (+) not allowed in operand of OR or IN

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

Table dropped.

```

Related Searches:

[Programming Using Java](#)

[Java Classes](#)

[Java Programming Tutorial](#)

[Java Programming Course](#)

[Application Development](#)



SQL>

Related Searches:

[Programming
Using Java](#)

[Java Classes](#)

[Java Programming
Tutorial](#)

[Java Programming
Course](#)

[Application
Development](#)

7.5.Outer Joins Limitations

7.5.1. [Limitations on Outer Joins](#)

7.5.2. You cannot use an outer join condition with the IN operator

7.5.3. [You cannot use an outer join condition with another join using the OR operator](#)

[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.

