

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

RevMob Mobile Ad Network

www.revmob.com

\$30 eCPMs, 100% Fill-Rates. Top ad network for iOS and Android!

AdChoices

7.1.7.Convert subqueries to JOINS

```
SQL>
SQL> CREATE TABLE emps (
  2   emp varchar(30)
  3   ,title   varchar(30)
  4   );

Table created.

SQL>
SQL> INSERT INTO emps VALUES ( 'Tom','Programmer' );

1 row created.

SQL> INSERT INTO emps VALUES ( 'Jack','Tester' );

1 row created.

SQL> INSERT INTO emps VALUES ( 'Mary','Technician' );

1 row created.

SQL>
SQL> CREATE TABLE JobLevel (
  2   title   varchar(30)
  3   ,rank    varchar(30)
  4   );

Table created.

SQL>
SQL> INSERT INTO JobLevel VALUES ( 'Programmer','Level1' );

1 row created.

SQL> INSERT INTO JobLevel VALUES ( 'Tester','Level2' );

1 row created.

SQL> INSERT INTO JobLevel VALUES ( 'Technician','Level3' );

1 row created.

SQL>
SQL> CREATE TABLE salary (
  2   rank    varchar(30)
  3   ,payment DECIMAL(10,2)
  4   );

Table created.

SQL>
SQL> INSERT INTO salary VALUES ( 'Level1',2000.00 );

1 row created.

SQL> INSERT INTO salary VALUES ( 'Level2',3000.00 );

1 row created.

SQL> INSERT INTO salary VALUES ( 'Level3',5000.00 );

1 row created.

SQL> INSERT INTO salary VALUES ( 'Level4',6000.00 );

1 row created.

SQL>
SQL> select * from emps;

EMP
-----
TITLE
-----
Tom
```

```

Programmer
Jack
Tester

Mary
Technician

3 rows selected.

SQL> select * from JobLevel;

TITLE
-----
RANK
-----
Programmer
Level1

Tester
Level2

Technician
Level3

3 rows selected.

SQL> select * from salary;

RANK                                PAYMENT
-----                                -
Level1                                2000
Level2                                3000
Level3                                5000
Level4                                6000

4 rows selected.

SQL>
SQL> SELECT payment FROM salary WHERE rank =
      2  (SELECT rank FROM JobLevel WHERE title =
      3  (SELECT title FROM emps WHERE emp = 'Jack'));

      PAYMENT
      -
      3000

1 row selected.

SQL>
SQL>
SQL>
SQL> DROP TABLE emps;

Table dropped.

SQL> DROP TABLE JobLevel;

Table dropped.

SQL> DROP TABLE salary;

Table dropped.

SQL>

```

[PI Sql Developer 8.0](http://www.allroundautomations.com)

www.allroundautomations.com

PS/SQL Developer! Order now or
download the trial version.



AdChoices

Make A Free Website



Click Here

WebStarts.com

7.1.Introduction

- 7.1.1. [Performing SELECT Statements that Use More than Two Tables](#)
- 7.1.2. [Three different types of joins:](#)
- 7.1.3. [Understanding Non-equijoins](#)
- 7.1.4. [Performing SELECT Statements that Use Two Tables](#)
- 7.1.5. [Example simple join.](#)
- 7.1.6. [Use table alias in table join](#)
- 7.1.7. [Convert subqueries to JOINS](#)
- 7.1.8. [autotrace ansi full outer join](#)

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.