**SQL Documentation**

**Join**

* Inner Join
* Left Join
* Right Join
* Full Join

**Set Operator**

* Union
* Union All
* Intersect
* Minus

**Functions**

* Character Functions

**Joins**

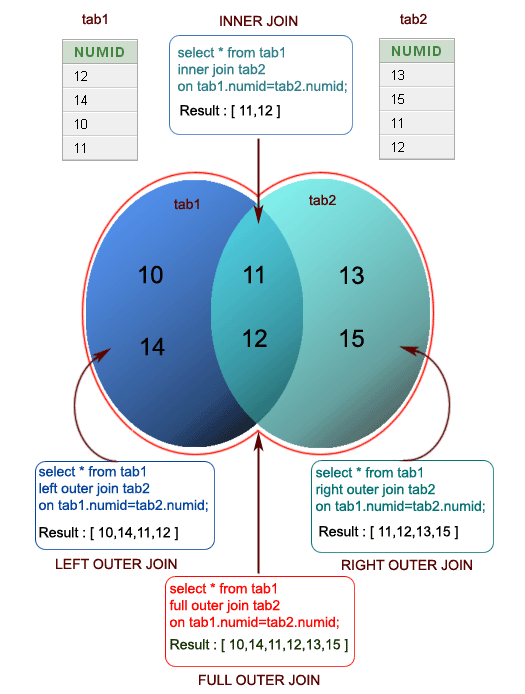


Table 1 - T\_NAME

|  |  |
| --- | --- |
| **Id** | **Name** |
| 1 | Ava |
| 2 | Ben |
| 3 | Cara |
| 4 | Dany |

Table 2 - T\_SALARY

|  |  |
| --- | --- |
| **Eid** | **Salary** |
| 2 | 1000 |
| 3 | 2000 |
| 5 | 3000 |
| 6 | 4000 |

Inner Join

SQL > SELECT ID, NAME, SALARY

FROM T\_NAME, T\_SALARY

WHERE T\_NAME.ID = T\_SALARY.EID

|  |  |  |
| --- | --- | --- |
| **Id** | **Name** | **Salary** |
| 2 | Ben | 1000 |
| 3 | Cara | 2000 |

**Left Join**

SQL > SELECT ID, NAME, SALARY

FROM T\_NAME, T\_SALARY

WHERE T\_NAME.ID = T\_SALARY.EID(+)

|  |  |  |
| --- | --- | --- |
| **Id** | **Name** | **Salary** |
| 1 | Ava | (null) |
| 2 | Ben | 1000 |
| 3 | Cara | 2000 |
| 4 | Dany | (null) |

**Right Join**

SQL > SELECT ID, NAME, SALARY

FROM T\_NAME, T\_SALARY

WHERE T\_NAME.ID(+) = T\_SALARY.EID

|  |  |  |
| --- | --- | --- |
| **Id** | **Name** | **Salary** |
| 2 | Ben | 1000 |
| 3 | Cara | 2000 |
| (null) | (null) | 3000 |
| (null) | (null) | 4000 |

**Full Outer Join**

SQL > SELECT ID, NAME, SALARY

FROM T\_NAME, T\_SALARY

WHERE T\_NAME.ID = T\_SALARY.EID(+)

UNION

SELECT ID, NAME, SALARY

FROM T\_NAME, T\_SALARY

WHERE T\_NAME.ID(+) = T\_SALARY.EID

|  |  |  |  |
| --- | --- | --- | --- |
| **Id** | **Name** | **Salary** | **Eid** |
| 1 | Ava | (null) | (null) |
| 2 | Ben | 1000 | 2 |
| 3 | Cara | 2000 | 3 |
| 4 | Dany | (null) | (null) |
| (null) | (null) | 3000 | 5 |
| (null) | (null) | 4000 | 6 |

**Set Operator**

Table 1 - TABLE\_A

|  |  |
| --- | --- |
| **Id** | **Name** |
| 1 | Ava |
| 2 | Ben |
| 3 | Cara |

Table 2 - TABLE\_B

|  |  |
| --- | --- |
| **Id** | **Name** |
| 3 | Cara |
| 4 | Dany |
| 5 | Edward |

**Union**

SQL > SELECT \* FROM TABLE\_A

UNION

SELECT \* FROM TABLE\_B

|  |  |
| --- | --- |
| **Id** | **Name** |
| 1 | Ava |
| 2 | Ben |
| 3 | Cara |
| 4 | Dany |
| 5 | Edward |

**Union All**

SQL > SELECT \* FROM TABLE\_A

UNION ALL

SELECT \* FROM TABLE\_B

|  |  |
| --- | --- |
| **Id** | **Name** |
| 1 | Ava |
| 2 | Ben |
| 3 | Cara |
| 3 | Cara |
| 4 | Dany |
| 5 | Edward |

**Intersect**

SQL > SELECT \* FROM TABLE\_A

INTERSECT

SELECT \* FROM TABLE\_B

|  |  |
| --- | --- |
| **Id** | **Name** |
| 3 | Cara |

**Minus**

SQL > SELECT \* FROM TABLE\_A

MINUS

SELECT \* FROM TABLE\_B

|  |  |
| --- | --- |
| **Id** | **Name** |
| 1 | Ava |
| 2 | Ben |

SQL > SELECT \* FROM TABLE\_B

MINUS

SELECT \* FROM TABLE\_A

|  |  |
| --- | --- |
| **Id** | **Name** |
| 4 | Dany |
| 5 | Edward |

**Character Functions**

**Case Manipulation Functions**

**Upper Case**

SQL > SELECT UPPER (‘Oracle Server’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| ORACLE SERVER |

**Lower Case**

SQL > SELECT LOWER(‘OrAcLe SeRvEr’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| oracle server |

**Initcap**

SQL > SELECT INITCAP(‘oracle server’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| Oracle Server |

SQL > SELECT INITCAP(‘a:b?c&d’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| A:B?C&D |

**String Manipulation Functions**

**Length**

SQL > SELECT LENGTH(‘Oracle Server’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| 13 |

**Concat**

SQL > SELECT CONCAT(‘Oracle’, ‘Server’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| OracleServer |

SQL > SELECT CONCAT(CONCAT(‘Oracle’, ‘Server’), ‘Link’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| OracleServerLink |

SQL > SELECT ‘Oracle’ || ‘Server’ || ‘Link’ AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| OracleServerLink |

**Lpad**

SQL > SELECT LPAD(‘Oracle’, 10, ‘\*’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| \*\*\*\*Oracle |

--> 4 (\*) + 6 (ch) = 10

SQL > SELECT LPAD(‘Oracle’, 15) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| Oracle |

--> 9 (space) + 6 (ch) = 15

**Rpad**

SQL > SELECT RPAD(‘Oracle’, 10, ‘$’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| Oracle$$$$ |

**Pad**

SQL > SELECT RPAD(LPAD(‘Oracle’, 10, \*), 14, ‘$’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| \*\*\*\*Oracle$$$$ |

**Ltrim**

SQL > SELECT LTRIM(‘Oracle’, ‘Or’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| acle |

SQL > SELECT LTRIM(‘oooooracle’, ‘o’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| racle |

**Rtrim**

SQL > SELECT RTRIM(‘Oracle’, ‘le’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| Orac |

SQL > SELECT RTRIM(‘Server’, ‘ver’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| S |

**Trim**

SQL > SELECT TRIM(‘o’ from ‘oooracleooo’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| racle |

SQL > SELECT TRIM(‘ oracle ’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| oracle |

SQL > SELECT TRIM(leading ‘O’ from ‘Oracle’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| racle |

|  |  |
| --- | --- |
| Trailing | Right |
| Both | Both |
| Leading | Left |

**Translate**

SQL > SELECT TRANSLATE(‘Oracle Server’, ‘OEL’, ‘123’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| 1rac32 S2rv2r |

SQL > SELECT TRANSLATE(‘Oracle Server’, ‘OEL’, ‘12’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| 1rac2 S2rv2r |

(L will be truncated)

SQL > SELECT TRANSLATE(‘Oracle Server’, ‘OEL’, ‘1234’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| 1rac32 S2rv2r |

(4 will not be displayed)

**Replace**

SQL > SELECT REPLACE(‘Oracle Server’, ‘OEL’, ‘er’, ‘12’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| Oracle S12v12 |

SQL > SELECT REPLACE(‘Oracle Server’, ‘OEL’, ‘er’, ‘1234’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| Oracle S1234v1234 |

**Substring**

SQL > SELECT SUBSTR(‘Oracle Server’, 1, 4) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| Orac |

--> 1 - start char, 4 - No.of char

SQL > SELECT SUBSTR(‘Oracle Server’, 4, 4) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| cle |

--> space is counted

SQL > SELECT SUBSTR(‘Oracle Server’, 4) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| cle Server |

SQL > SELECT SUBSTR(‘Oracle Server’, -1) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| r |

--> -1 - counted from reverse

SQL > SELECT SUBSTR(‘Oracle Server’, -9, 4) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| le S |

--> -ve will start from right side and towards right side

**Instr**

SQL > SELECT INSTR(‘Oracle Server’, ‘r’, 1, 1) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| 2 |

--> r - Search

1 - Start position

1 - Occurence

2 - Position

SQL > SELECT INSTR(‘Oracle Server’, ‘e’, 1, 3) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| 12 |

SQL > SELECT INSTR(‘Oracle Server’, ‘e’, 7, 3) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| 0 |

--> 3rd Occurence not in the string after 7th position.

SQL > SELECT INSTR(‘Oracle Server’, ‘e’, 7) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| 9 |

--> 7 - start

SQL > SELECT INSTR(‘Oracle Server’, ‘e’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| 6 |

SQL > SELECT INSTR(‘Oracle Server’, ‘e’, -1) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| 12 |

--> Checking right to left

Position left to right

**Reverse**

SQL > SELECT REVERSE(‘Oracle’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| elcarO |

SQL > SELECT REVERSE(‘Oracle Server’) AS OUTPUT FROM DUAL

|  |
| --- |
| **Output** |
| revreS elcarO |

END