무료 전자 책

배우기

Oracle Database

Free unaffiliated eBook created from
Stack Overflow contributors.

#oracle

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: oracle-database

It is an unofficial and free Oracle Database ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official Oracle Database.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

1: Oracle Database

70 Larry Ellison, Bob Miner Ed Oates (RDBMS). IBM System R .

1 ( )	1978-01-01
Oracle V2	1979-01-01
Oracle 3	1983-01-01
Oracle 4	1984-01-01
Oracle 5	1985-01-01
Oracle 6	1988-01-01
Oracle7	1992-01-01
Oracle8	1997-07-01
Oracle8i	1999-02-01
Oracle9i	2001-06-01
10g	2003-01-01
11g	2007-01-01
12c	2013-01-01

Examples

```
SELECT 'Hello world!' FROM dual;
```

SQL " convienence table" . JOIN `DUMMY` 'X' .

!

```
create table MY_table (
   what varchar2(10),
   who varchar2(10),
   mark varchar2(10)
);
```

( )

```
insert into my_table (what, who, mark) values ('Hello', 'world', '!' );
insert into my_table values ('Bye bye', 'ponies', '?' );
insert into my_table (what) values('Hey');
```

Oracle

```
commit;
```

:

```
select what, who, mark from my_table where what='Hello';
```

SQL

$ 50000 . , .

```
SELECT employee_name, date_of_birth, salary
FROM   employees
WHERE  salary > 50000
   AND date_of_birth >= DATE '2000-01-01'
ORDER BY employee_name;
```

5 . .

```
SELECT department_id, COUNT(*)
FROM   employees
GROUP BY department_id
HAVING COUNT(*) >= 5
ORDER BY COUNT(*) DESC;
```

PL / SQL Hello World

```
/* PL/SQL is a core Oracle Database technology, allowing you to build clean, secure,
   optimized APIs to SQL and business logic. */

set serveroutput on

BEGIN
   DBMS_OUTPUT.PUT_LINE ('Hello World!');
END;
```

Oracle Database : https://riptutorial.com/ko/oracle/topic/558/oracle-database-

2: NULL

NULL. NULL `a = NULL` . `a IS NULL` `a IS NOT NULL` . NULL NULL . null . NULL NULL . , `3 * NULL + 5` .

PRIMARY KEY NOT NULL NULL . (NOVALIDATE)

Examples

NULL .

```
SELECT 1 NUM_COLUMN, 'foo' VARCHAR2_COLUMN from DUAL
UNION ALL
SELECT NULL, NULL from DUAL;
```

NUM_COLUMN	VARCHAR2_COLUMN
1	
()	()

NULL.

```
SELECT 1 a, '' b from DUAL;
```

1	()

NULL NULL.

```
SELECT 3 * NULL + 5, 'Hello ' || NULL || 'world'   from DUAL;
```

3 * NULL + 5	'HELLO'|| NULL || 'WORLD'
()	

null NVL

```
SELECT a column_with_null, NVL(a, 'N/A') column_without_null FROM
  (SELECT NULL a FROM DUAL);
```

COLUMN_WITH_NULL	COLUMN_WITHOUT_NULL
()	N / A

NVL NULL .

```
SELECT
    CASE WHEN a = b THEN 1 WHEN a <> b THEN 0 else -1 END comparison_without_nvl,
    CASE WHEN NVL(a, -1) = NVL(b, -1) THEN 1 WHEN NVL(a, -1) <> NVL(b, -1) THEN 0 else -1 END
comparison_with_nvl
  FROM
    (select null a, 3 b FROM DUAL
     UNION ALL
     SELECT NULL, NULL FROM DUAL);
```

COMPARISON_WITHOUT_NVL	COMPARISON_WITH_NVL
-1	0
-1	1

(null) NVL2 .

NOT NULL NVL2 . .

```
SELECT NVL2(null, 'Foo', 'Bar'), NVL2(5, 'Foo', 'Bar') FROM DUAL;
```

NVL2 (NULL, 'FOO', 'BAR')	NVL2 (5, 'FOO', 'BAR')

NULL COALESCE

```
SELECT COALESCE(a, b, c, d, 5) FROM
    (SELECT NULL A, NULL b, NULL c, 4 d FROM DUAL);
```

(A, B, C, D, 5)
4

COALESCE NVL . NVL . COALESCE NULL . , NULL COALESCE .

NULL : https://riptutorial.com/ko/oracle/topic/8183/null--

3: Oracle Advanced Queuing (AQ)

- `dbms_aqadm.create_queue_table` DDL DML . dbms_aqadm dbms_aq . , . DDL DML Oracle Support .
- `dbms_aq.forever` . (Oracle Doc ID 2001165.1).
- 10.1 AQ_TM_PROCESSES . QMON 0 . Oracle . `alter system reset aq_tm_processes scope=spfile sid='*';`

Examples

/

. . , .

Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit Production .

, . . AQ 3600 . 48 .

```
create type message_t as object
   (
   sender  varchar2 ( 50 ),
   message varchar2 ( 512 )
   );
/
-- Type MESSAGE_T compiled
begin dbms_aqadm.create_queue_table(
     queue_table        => 'MESSAGE_Q_TBL',
     queue_payload_type => 'MESSAGE_T',
     sort_list          => 'PRIORITY,ENQ_TIME',
     multiple_consumers =>  false,
     compatible         => '10.0.0');
  end;
/
-- PL/SQL procedure successfully completed.
begin dbms_aqadm.create_queue(
     queue_name          => 'MESSAGE_Q',
     queue_table         => 'MESSAGE_Q_TBL',
     queue_type          =>  0,
     max_retries         =>  48,
     retry_delay         =>  3600,
     dependency_tracking =>  false);
  end;
/
-- PL/SQL procedure successfully completed.
```

.

```
create or replace package message_worker_pkg
is
   queue_name_c constant varchar2(20) := 'MESSAGE_Q';
```

```
   -- allows the workers to process messages in the queue
   procedure enable_dequeue;

   -- prevents messages from being worked but will still allow them to be created and enqueued
   procedure disable_dequeue;

   -- called only by Oracle Advanced Queueing.  Do not call anywhere else.
   procedure on_message_enqueued (context        in raw,
                                  reginfo        in sys.aq$_reg_info,
                                  descr          in sys.aq$_descriptor,
                                  payload        in raw,
                                  payloadl       in number);

   -- allows messages to be worked if we missed the notification (or a retry
   -- is pending)
   procedure work_old_messages;

end;
/

create or replace package body message_worker_pkg
is
   -- raised by Oracle when we try to dequeue but no more messages are ready to
   -- be dequeued at this moment
   no_more_messages_ex          exception;
   pragma exception_init (no_more_messages_ex,
                          -25228);

   -- allows the workers to process messages in the queue
   procedure enable_dequeue
   as
   begin
      dbms_aqadm.start_queue (queue_name => queue_name_c, dequeue => true);
   end enable_dequeue;

   -- prevents messages from being worked but will still allow them to be created and enqueued
   procedure disable_dequeue
   as
   begin
      dbms_aqadm.stop_queue (queue_name => queue_name_c, dequeue => true, enqueue => false);
   end disable_dequeue;

   procedure work_message (message_in in out nocopy message_t)
   as
   begin
      dbms_output.put_line ( message_in.sender || ' says ' || message_in.message );
   end work_message;

   -- called only by Oracle Advanced Queueing.  Do not call anywhere else.

   procedure on_message_enqueued (context        in raw,
                                  reginfo        in sys.aq$_reg_info,
                                  descr          in sys.aq$_descriptor,
                                  payload        in raw,
                                  payloadl       in number)
   as
      pragma autonomous_transaction;
      dequeue_options_l      dbms_aq.dequeue_options_t;
      message_id_l           raw (16);
      message_l              message_t;
      message_properties_l   dbms_aq.message_properties_t;
```

```
   begin
      dequeue_options_l.msgid         := descr.msg_id;
      dequeue_options_l.consumer_name := descr.consumer_name;
      dequeue_options_l.wait          := dbms_aq.no_wait;
      dbms_aq.dequeue (queue_name           => descr.queue_name,
                       dequeue_options      => dequeue_options_l,
                       message_properties   => message_properties_l,
                       payload              => message_l,
                       msgid                => message_id_l);
      work_message (message_l);
      commit;
   exception
      when no_more_messages_ex
      then
         -- it's possible work_old_messages already dequeued the message
         commit;
      when others
      then
         -- we don't need to have a raise here.  I just wanted to point out that
         -- since this will be called by AQ throwing the exception back to it
         -- will have it put the message back on the queue and retry later
         raise;
   end on_message_enqueued;

   -- allows messages to be worked if we missed the notification (or a retry
   -- is pending)
   procedure work_old_messages
   as
      pragma autonomous_transaction;
      dequeue_options_l      dbms_aq.dequeue_options_t;
      message_id_l           raw (16);
      message_l              message_t;
      message_properties_l   dbms_aq.message_properties_t;
   begin
      dequeue_options_l.wait       := dbms_aq.no_wait;
      dequeue_options_l.navigation := dbms_aq.first_message;

      while (true) loop -- way out is no_more_messages_ex
         dbms_aq.dequeue (queue_name           => queue_name_c,
                          dequeue_options      => dequeue_options_l,
                          message_properties   => message_properties_l,
                          payload              => message_l,
                          msgid                => message_id_l);
         work_message (message_l);
         commit;
      end loop;
   exception
      when no_more_messages_ex
      then
         null;
   end work_old_messages;
end;
```

MESSAGE_Q ( ) AQ . AQ .

```
begin
  dbms_aq.register (
     sys.aq$_reg_info_list (
        sys.aq$_reg_info (user || '.' || message_worker_pkg.queue_name_c,
                          dbms_aq.namespace_aq,
```

```
                          'plsql://' || user || '.message_worker_pkg.on_message_enqueued',
                          hextoraw ('FF'))),
     1);
  commit;
end;
```

```
declare
   enqueue_options_l      dbms_aq.enqueue_options_t;
   message_properties_l   dbms_aq.message_properties_t;
   message_id_l           raw (16);
   message_l              message_t;
begin
   -- only need to do this next line ONCE
   dbms_aqadm.start_queue (queue_name => message_worker_pkg.queue_name_c, enqueue => true ,
dequeue => true);

   message_l := new message_t ( 'Jon', 'Hello, world!' );
   dbms_aq.enqueue (queue_name           => message_worker_pkg.queue_name_c,
                    enqueue_options      => enqueue_options_l,
                    message_properties   => message_properties_l,
                    payload              => message_l,
                    msgid                => message_id_l);
   commit;
end;
```

Oracle Advanced Queuing (AQ) : https://riptutorial.com/ko/oracle/topic/4362/oracle-advanced-queuing--aq-

4: Oracle Database 12C

Examples

Caluse

```
SELECT E.EMPLOYEE_ID,E.LAST_NAME,E.MANAGER_ID FROM HR.EMPLOYEES E
CONNECT BY PRIOR E.EMPLOYEE_ID = E.MANAGER_ID;
```

CONNECT BY .

```
SELECT E.LAST_NAME|| ' reports to ' ||
PRIOR E.LAST_NAME "Walk Top Down"
FROM HR.EMPLOYEES E
START WITH E.MANAGER_ID IS NULL
CONNECT BY PRIOR E.EMPLOYEE_ID = E.MANAGER_ID;
```

Oracle Database 12C : https://riptutorial.com/ko/oracle/topic/8777/oracle-database-12c----

5: WITH (AKA)

Oracle 11g R2 .

Examples

:

```
WITH generator ( value ) AS (
  SELECT 1 FROM DUAL
UNION ALL
  SELECT value + 1
  FROM   generator
  WHERE  value < 10
)
SELECT value
FROM   generator;
```

:

```
VALUE
-----
    1
    2
    3
    4
    5
    6
    7
    8
    9
   10
```

:

```
CREATE TABLE table_name ( value VARCHAR2(50) );

INSERT INTO table_name ( value ) VALUES ( 'A,B,C,D,E' );
```

:

```
WITH items ( list, item, lvl ) AS (
  SELECT value,
         REGEXP_SUBSTR( value, '[^,]+', 1, 1 ),
         1
  FROM   table_name
UNION ALL
  SELECT value,
         REGEXP_SUBSTR( value, '[^,]+', 1, lvl + 1 ),
         lvl + 1
  FROM   items
  WHERE  lvl < REGEXP_COUNT( value, '[^,]+' )
```

```
)
SELECT * FROM items;
```

:

```
LIST      ITEM LVL
--------- ---- ---
A,B,C,D,E    A   1
A,B,C,D,E    B   2
A,B,C,D,E    C   3
A,B,C,D,E    D   4
A,B,C,D,E    E   5
```

WITH (AKA) : https://riptutorial.com/ko/oracle/topic/3506/with--------aka----

6:

Examples

CROSS JOIN . . TABLEA 20 TABLEB 20 20*20 = 400 .

:

```
SELECT *
FROM TABLEA CROSS JOIN TABLEB;
```

:

```
SELECT *
FROM TABLEA, TABLEB;
```

SQL .

: TABLEA

```
+-------+---------+
| VALUE |   NAME  |
+-------+---------+
|   1   |   ONE   |
|   2   |   TWO   |
+-------+---------+
```

: TABLEB

```
+-------+--------+
| VALUE |  NAME  |
+-------+--------+
|   3   |  THREE |
|   4   |  FOUR  |
+-------+--------+
```

, :

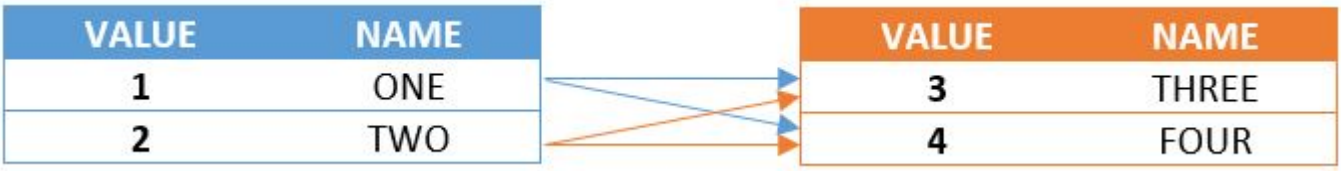
```
SELECT *
FROM TABLEA CROSS JOIN TABLEB;
```

:

```
+-------+--------+-------+--------+
| VALUE |  NAME  | VALUE |  NAME  |
+-------+--------+-------+--------+
|   1   |   ONE  |   3   |  THREE |
|   1   |   ONE  |   4   |  FOUR  |
|   2   |   TWO  |   3   |  THREE |
|   2   |   TWO  |   4   |  FOUR  |
```

```
+-------+--------+-------+--------+
```

.



Cross Join : Oracle

INNER JOIN JOIN .

TableExpression [INNER] JOIN TableExpression {ON booleanExpression | USING }

ON .

ON SELECT . ON .

```
-- Join the EMP_ACT and EMPLOYEE tables
-- select all the columns from the EMP_ACT table and
-- add the employee's surname (LASTNAME) from the EMPLOYEE table
-- to each row of the result
SELECT SAMP.EMP_ACT.*, LASTNAME
 FROM SAMP.EMP_ACT JOIN SAMP.EMPLOYEE
 ON EMP_ACT.EMPNO = EMPLOYEE.EMPNO
-- Join the EMPLOYEE and DEPARTMENT tables,
-- select the employee number (EMPNO),
-- employee surname (LASTNAME),
-- department number (WORKDEPT in the EMPLOYEE table and DEPTNO in the
-- DEPARTMENT table)
-- and department name (DEPTNAME)
-- of all employees who were born (BIRTHDATE) earlier than 1930.
SELECT EMPNO, LASTNAME, WORKDEPT, DEPTNAME
 FROM SAMP.EMPLOYEE JOIN SAMP.DEPARTMENT
 ON WORKDEPT = DEPTNO
 AND YEAR(BIRTHDATE) < 1930

-- Another example of "generating" new data values,
-- using a query which selects from a VALUES clause (which is an
-- alternate form of a fullselect).
-- This query shows how a table can be derived called "X"
-- having 2 columns "R1" and "R2" and 1 row of data
SELECT *
FROM (VALUES (3, 4), (1, 5), (2, 6))
AS VALUESTABLE1(C1, C2)
JOIN (VALUES (3, 2), (1, 2),
(0, 3)) AS VALUESTABLE2(c1, c2)
ON VALUESTABLE1.c1 = VALUESTABLE2.c1
-- This results in:
-- C1          |C2          |C1          |2
-- -----------------------------------------------
-- 3           |4           |3           |2
-- 1           |5           |1           |2
```

```
-- List every department with the employee number and
-- last name of the manager

SELECT DEPTNO, DEPTNAME, EMPNO, LASTNAME
FROM DEPARTMENT INNER JOIN EMPLOYEE
ON MGRNO = EMPNO

-- List every employee number and last name
-- with the employee number and last name of their manager
SELECT E.EMPNO, E.LASTNAME, M.EMPNO, M.LASTNAME
FROM EMPLOYEE E INNER JOIN
DEPARTMENT INNER JOIN EMPLOYEE M
    ON MGRNO = M.EMPNO
    ON E.WORKDEPT = DEPTNO
```

LEFT OUTER JOIN

`LEFT OUTER JOIN` .

:

```
SELECT
      ENAME,
      DNAME,
      EMP.DEPTNO,
      DEPT.DEPTNO
FROM
      SCOTT.EMP LEFT OUTER JOIN SCOTT.DEPT
      ON EMP.DEPTNO = DEPT.DEPTNO;
```

ANSI . `(+)` .

```
SELECT
      ENAME,
      DNAME,
      EMP.DEPTNO,
      DEPT.DEPTNO
FROM
      SCOTT.EMP,
      SCOTT.DEPT
WHERE
      EMP.DEPTNO = DEPT.DEPTNO(+);
```

.

: EMPLOYEE

```
+-----------+---------+
|    NAME   |  DEPTNO |
+-----------+---------+
|     A     |    2    |
|     B     |    1    |
|     C     |    3    |
|     D     |    2    |
|     E     |    1    |
|     F     |    1    |
```

```
|     G     |    4    |
|     H     |    4    |
+-----------+---------+
```

: DEPT

```
+---------+--------------+
|  DEPTNO |   DEPTNAME   |
+---------+--------------+
|    1    |  ACCOUNTING  |
|    2    |    FINANCE   |
|    5    |   MARKETING  |
|    6    |      HR      |
+---------+--------------+
```

, :

```
SELECT
      *
FROM
      EMPLOYEE LEFT OUTER JOIN DEPT
      ON EMPLOYEE.DEPTNO = DEPT.DEPTNO;
```

:

```
+-----------+---------+---------+--------------+
|    NAME   |  DEPTNO |  DEPTNO |   DEPTNAME   |
+-----------+---------+---------+--------------+
|     F     |    1    |    1    |  ACCOUNTING  |
|     E     |    1    |    1    |  ACCOUNTING  |
|     B     |    1    |    1    |  ACCOUNTING  |
|     D     |    2    |    2    |    FINANCE   |
|     A     |    2    |    2    |    FINANCE   |
|     C     |    3    |         |              |
|     H     |    4    |         |              |
|     G     |    4    |         |              |
+-----------+---------+---------+--------------+
```

RIGHT OUTER JOIN .

:

```
SELECT
      ENAME,
      DNAME,
      EMP.DEPTNO,
      DEPT.DEPTNO
FROM
      SCOTT.EMP RIGHT OUTER JOIN SCOTT.DEPT
      ON EMP.DEPTNO = DEPT.DEPTNO;
```

SCOTT.DEPT SCOTT.DEPT SCOTT.EMP LEFT OUTER JOIN .

```
SELECT
```

```
      ENAME,
      DNAME,
      EMP.DEPTNO,
      DEPT.DEPTNO
FROM
      SCOTT.DEPT RIGHT OUTER JOIN SCOTT.EMP
      ON DEPT.DEPTNO = EMP.DEPTNO;
```

.

: EMPLOYEE

```
+-----------+---------+
|    NAME   |  DEPTNO |
+-----------+---------+
|     A     |    2    |
|     B     |    1    |
|     C     |    3    |
|     D     |    2    |
|     E     |    1    |
|     F     |    1    |
|     G     |    4    |
|     H     |    4    |
+-----------+---------+
```

: DEPT

```
+---------+--------------+
|  DEPTNO |   DEPTNAME   |
+---------+--------------+
|    1    |  ACCOUNTING  |
|    2    |    FINANCE   |
|    5    |   MARKETING  |
|    6    |      HR      |
+---------+--------------+
```

, :

```
SELECT
      *
FROM
      EMPLOYEE RIGHT OUTER JOIN DEPT
      ON EMPLOYEE.DEPTNO = DEPT.DEPTNO;
```

:

```
+-----------+---------+---------+--------------+
|    NAME   |  DEPTNO |  DEPTNO |   DEPTNAME   |
+-----------+---------+---------+--------------+
|     A     |    2    |    2    |    FINANCE   |
|     B     |    1    |    1    |  ACCOUNTING  |
|     D     |    2    |    2    |    FINANCE   |
|     E     |    1    |    1    |  ACCOUNTING  |
|     F     |    1    |    1    |  ACCOUNTING  |
|           |         |    5    |   MARKETING  |
|           |         |    6    |      HR      |
```

```
+-----------+---------+---------+--------------+
```

Oracle (+) .

```
SELECT *
FROM EMPLOYEE, DEPT
WHERE EMPLOYEE.DEPTNO(+) = DEPT.DEPTNO;
```

```
FULL OUTER JOIN           .,   .
```

:

```
SELECT
      *
FROM
      EMPLOYEE FULL OUTER JOIN DEPT
      ON EMPLOYEE.DEPTNO = DEPT.DEPTNO;
```

.

: EMPLOYEE

```
+-----------+---------+
|    NAME   |  DEPTNO |
+-----------+---------+
|     A     |    2    |
|     B     |    1    |
|     C     |    3    |
|     D     |    2    |
|     E     |    1    |
|     F     |    1    |
|     G     |    4    |
|     H     |    4    |
+-----------+---------+
```

: DEPT

```
+---------+--------------+
|  DEPTNO |   DEPTNAME   |
+---------+--------------+
|    1    |  ACCOUNTING  |
|    2    |    FINANCE   |
|    5    |   MARKETING  |
|    6    |      HR      |
+---------+--------------+
```

, :

```
SELECT
      *
FROM
      EMPLOYEE FULL OUTER JOIN DEPT
      ON EMPLOYEE.DEPTNO = DEPT.DEPTNO;
```

```
+-----------+---------+---------+--------------+
|    NAME   |  DEPTNO |  DEPTNO |   DEPTNAME   |
+-----------+---------+---------+--------------+
|     A     |    2    |    2    |    FINANCE   |
|     B     |    1    |    1    |  ACCOUNTING  |
|     C     |    3    |         |              |
|     D     |    2    |    2    |    FINANCE   |
|     E     |    1    |    1    |  ACCOUNTING  |
|     F     |    1    |    1    |  ACCOUNTING  |
|     G     |    4    |         |              |
|     H     |    4    |         |              |
|           |         |    6    |      HR      |
|           |         |    5    |   MARKETING  |
+-----------+---------+---------+--------------+
```

NULL .

antijoin . (NOT IN).

```
SELECT * FROM employees
   WHERE department_id NOT IN
   (SELECT department_id FROM departments
       WHERE location_id = 1700)
   ORDER BY last_name;
```

.

: EMPLOYEE

```
+-----------+---------+
|    NAME   |  DEPTNO |
+-----------+---------+
|     A     |    2    |
|     B     |    1    |
|     C     |    3    |
|     D     |    2    |
|     E     |    1    |
|     F     |    1    |
|     G     |    4    |
|     H     |    4    |
+-----------+---------+
```

: DEPT

```
+---------+--------------+
|  DEPTNO |   DEPTNAME   |
+---------+--------------+
|    1    |  ACCOUNTING  |
|    2    |    FINANCE   |
|    5    |   MARKETING  |
|    6    |      HR      |
+---------+--------------+
```

, :

```
SELECT
      *
FROM
      EMPLOYEE WHERE DEPTNO NOT IN
      (SELECT DEPTNO FROM DEPT);
```

:

```
+-----------+---------+
|    NAME   |  DEPTNO |
+-----------+---------+
|     C     |    3    |
|     H     |    4    |
|     G     |    4    |
+-----------+---------+
```

EEPLOYEE DEPTNO DEPT .

SEMIJOIN

(semijoin) 2500 .

```
SELECT * FROM departments
   WHERE EXISTS
   (SELECT 1 FROM employees
       WHERE departments.department_id = employees.department_id
       AND employees.salary > 2500)
   ORDER BY department_name;
```

. 2500 where . `n` 3000, `select * from departments, employees` ID where `n` .

`JOIN` x . Oracle 9i `JOIN` `INNER JOIN` . `CROSS JOIN` `NATURAL JOIN` .

:

```
select t1.*,
       t2.DeptId
  from table_1 t1
  join table_2 t2 on t2.DeptNo = t1.DeptNo
```

:

- 10g
- 11g
- 12g

NATURAL JOIN

NATURAL JOIN . .

```
create table tab1(id number,  descr varchar2(100));
create table tab2(id number,  descr varchar2(100));
```

```
insert into tab1 values(1, 'one');
insert into tab1 values(2, 'two');
insert into tab1 values(3, 'three');
insert into tab2 values(1, 'ONE');
insert into tab2 values(3, 'three');
```

ID DESCR .

```
SQL> select *
  2  from tab1
  3        natural join
  4       tab2;

        ID DESCR
---------- ----------
         3 three
```

JOIN .

```
SQL> select *
  2  from (select id as id, descr as descr1 from tab1)
  3       natural join
  4       (select id as id, descr as descr2 from tab2);

        ID DESCR1     DESCR2
---------- ---------- ----------
         1 one        ONE
         3 three      three
```

JOIN .

```
SQL> select *
  2  from (select id as id1, descr as descr1 from tab1)
  3       natural join
  4       (select id as id2, descr as descr2 from tab2);

       ID1 DESCR1            ID2 DESCR2
---------- ---------- ---------- ----------
         1 one                 1 ONE
         2 two                 1 ONE
         3 three               1 ONE
         1 one                 3 three
         2 two                 3 three
         3 three               3 three
```

: https://riptutorial.com/ko/oracle/topic/4192/

7:

Examples

:

```
CREATE TABLE table_name ( id, list ) AS
SELECT 1, 'a,b,c,d' FROM DUAL UNION ALL -- Multiple items in the list
SELECT 2, 'e'       FROM DUAL UNION ALL -- Single item in the list
SELECT 3, NULL      FROM DUAL UNION ALL -- NULL list
SELECT 4, 'f,,g'    FROM DUAL;          -- NULL item in the list
```

:

```
WITH bounds ( id, list, start_pos, end_pos, lvl ) AS (
  SELECT id, list, 1, INSTR( list, ',' ), 1 FROM table_name
UNION ALL
  SELECT id,
         list,
         end_pos + 1,
         INSTR( list, ',', end_pos + 1 ),
         lvl + 1
  FROM   bounds
  WHERE  end_pos > 0
)
SELECT id,
       SUBSTR(
         list,
         start_pos,
         CASE end_pos
           WHEN 0
           THEN LENGTH( list ) + 1
           ELSE end_pos
         END - start_pos
       ) AS item,
      lvl
FROM   bounds
ORDER BY id, lvl;
```

:

```
        ID ITEM           LVL
---------- ------- ----------
         1 a                1
         1 b                2
         1 c                3
         1 d                4
         2 e                1
         3 (NULL)           1
         4 f                1
         4 (NULL)           2
         4 g                3
```

PL / SQL

PL / SQL :

```
CREATE OR REPLACE FUNCTION split_String(
  i_str    IN  VARCHAR2,
  i_delim  IN  VARCHAR2 DEFAULT ','
) RETURN SYS.ODCIVARCHAR2LIST DETERMINISTIC
AS
  p_result       SYS.ODCIVARCHAR2LIST := SYS.ODCIVARCHAR2LIST();
  p_start        NUMBER(5) := 1;
  p_end          NUMBER(5);
  c_len CONSTANT NUMBER(5) := LENGTH( i_str );
  c_ld  CONSTANT NUMBER(5) := LENGTH( i_delim );
BEGIN
  IF c_len > 0 THEN
    p_end := INSTR( i_str, i_delim, p_start );
    WHILE p_end > 0 LOOP
      p_result.EXTEND;
      p_result( p_result.COUNT ) := SUBSTR( i_str, p_start, p_end - p_start );
      p_start := p_end + c_ld;
      p_end := INSTR( i_str, i_delim, p_start );
    END LOOP;
    IF p_start <= c_len + 1 THEN
      p_result.EXTEND;
      p_result( p_result.COUNT ) := SUBSTR( i_str, p_start, c_len - p_start + 1 );
    END IF;
  END IF;
  RETURN p_result;
END;
/
```

:

```
CREATE TABLE table_name ( id, list ) AS
SELECT 1, 'a,b,c,d' FROM DUAL UNION ALL -- Multiple items in the list
SELECT 2, 'e'       FROM DUAL UNION ALL -- Single item in the list
SELECT 3, NULL      FROM DUAL UNION ALL -- NULL list
SELECT 4, 'f,,g'    FROM DUAL;          -- NULL item in the list
```

:

```
SELECT t.id,
       v.column_value AS value,
       ROW_NUMBER() OVER ( PARTITION BY id ORDER BY ROWNUM ) AS lvl
FROM   table_name t,
       TABLE( split_String( t.list ) ) (+) v
```

:

```
        ID ITEM           LVL
---------- ------- ----------
         1 a                1
         1 b                2
         1 c                3
         1 d                4
```

```
         2 e                1
         3 (NULL)           1
         4 f                1
         4 (NULL)           2
         4 g                3
```

:

```
CREATE TABLE table_name ( id, list ) AS
SELECT 1, 'a,b,c,d' FROM DUAL UNION ALL -- Multiple items in the list
SELECT 2, 'e'       FROM DUAL UNION ALL -- Single item in the list
SELECT 3, NULL      FROM DUAL UNION ALL -- NULL list
SELECT 4, 'f,,g'    FROM DUAL;          -- NULL item in the list
```

:

```
SELECT t.id,
       v.COLUMN_VALUE AS value,
       ROW_NUMBER() OVER ( PARTITION BY id ORDER BY ROWNUM ) AS lvl
FROM   table_name t,
       TABLE(
         CAST(
           MULTISET(
             SELECT REGEXP_SUBSTR( t.list, '([^,]*)(,|$)', 1, LEVEL, NULL, 1 )
             FROM   DUAL
             CONNECT BY LEVEL < REGEXP_COUNT( t.list, '[^,]*(,|$)' )
           )
           AS SYS.ODCIVARCHAR2LIST
         )
       ) v;
```

:

```
        ID ITEM           LVL
---------- ------- ----------
         1 a                1
         1 b                2
         1 c                3
         1 d                4
         2 e                1
         3 (NULL)           1
         4 f                1
         4 (NULL)           2
         4 g                3
```

:

```
CREATE TABLE table_name ( id, list ) AS
SELECT 1, 'a,b,c,d' FROM DUAL UNION ALL -- Multiple items in the list
SELECT 2, 'e'       FROM DUAL UNION ALL -- Single item in the list
SELECT 3, NULL      FROM DUAL UNION ALL -- NULL list
SELECT 4, 'f,,g'    FROM DUAL;          -- NULL item in the list
```

:

```
SELECT t.id,
       REGEXP_SUBSTR( list, '([^,]*)(,|$)', 1, LEVEL, NULL, 1 ) AS value,
       LEVEL AS lvl
FROM   table_name t
CONNECT BY
       id = PRIOR id
AND    PRIOR SYS_GUID() IS NOT NULL
AND    LEVEL < REGEXP_COUNT( list, '([^,]*)(,|$)' )
```

:

```
        ID ITEM           LVL
---------- ------- ----------
         1 a                1
         1 b                2
         1 c                3
         1 d                4
         2 e                1
         3 (NULL)           1
         4 f                1
         4 (NULL)           2
         4 g                3
```

XMLTable FLWOR

Oracle 11 `ora:tokenize` XQuery .

:

```
CREATE TABLE table_name ( id, list ) AS
SELECT 1, 'a,b,c,d' FROM DUAL UNION ALL -- Multiple items in the list
SELECT 2, 'e'       FROM DUAL UNION ALL -- Single item in the list
SELECT 3, NULL      FROM DUAL UNION ALL -- NULL list
SELECT 4, 'f,,g'    FROM DUAL;          -- NULL item in the list
```

:

```
SELECT t.id,
       x.item,
       x.lvl
FROM   table_name t,
       XMLTABLE(
         'let $list := ora:tokenize(.,","),
              $cnt := count($list)
          for $val at $r in $list
          where $r < $cnt
          return $val'
       PASSING list||','
       COLUMNS
         item VARCHAR2(100) PATH '.',
         lvl FOR ORDINALITY
       ) (+) x;
```

:

```
        ID ITEM           LVL
---------- ------- ----------
         1 a                1
         1 b                2
         1 c                3
         1 d                4
         2 e                1
         3 (NULL)       (NULL)
         4 f                1
         4 (NULL)           2
         4 g                3
```

CROSS APPLY (Oracle 12c)

:

```
CREATE TABLE table_name ( id, list ) AS
SELECT 1, 'a,b,c,d' FROM DUAL UNION ALL -- Multiple items in the list
SELECT 2, 'e'       FROM DUAL UNION ALL -- Single item in the list
SELECT 3, NULL      FROM DUAL UNION ALL -- NULL list
SELECT 4, 'f,,g'    FROM DUAL;          -- NULL item in the list
```

:

```
SELECT t.id,
       REGEXP_SUBSTR( t.list, '([^,]*)($|,)', 1, l.lvl, NULL, 1 ) AS item,
       l.lvl
FROM   table_name t
       CROSS APPLY
       (
         SELECT LEVEL AS lvl
         FROM   DUAL
         CONNECT BY LEVEL <= REGEXP_COUNT( t.list, ',' ) + 1
       ) l;
```

:

```
        ID ITEM           LVL
---------- ------- ----------
         1 a                1
         1 b                2
         1 c                3
         1 d                4
         2 e                1
         3 (NULL)           1
         4 f                1
         4 (NULL)           2
         4 g                3
```

XMLTable

:

```
CREATE TABLE table_name ( id, list ) AS
```

```
SELECT 1, 'a,b,c,d' FROM DUAL UNION ALL -- Multiple items in the list
SELECT 2, 'e'       FROM DUAL UNION ALL -- Single item in the list
SELECT 3, NULL      FROM DUAL UNION ALL -- NULL list
SELECT 4, 'f,,g'    FROM DUAL;          -- NULL item in the list
```

:

```
SELECT t.id,
       SUBSTR( x.item.getStringVal(), 2 ) AS item,
       x.lvl
FROM   table_name t
       CROSS JOIN
       XMLTABLE(
         ( '"#' || REPLACE( t.list, ',', '","#' ) || '"' )
         COLUMNS item XMLTYPE PATH '.',
                 lvl  FOR ORDINALITY
       ) x;
```

(: # NULL , SUBSTR(item, 2) . NULL .)

:

```
        ID ITEM            LVL
---------- ------- ----------
         1 a                 1
         1 b                 2
         1 c                 3
         1 d                 4
         2 e                 1
         3 (NULL)            1
         4 f                 1
         4 (NULL)            2
         4 g                 3
```

: https://riptutorial.com/ko/oracle/topic/1968/---

8:

Examples

DATE . / / 0 (:).

ANSI DATE (ISO 8601).

```
SELECT DATE '2000-01-01' FROM DUAL;
```

TO_DATE() :

```
SELECT TO_DATE( '2001-01-01', 'YYYY-MM-DD' ) FROM DUAL;
```

(Oracle .)

:

```
SELECT TO_DATE(
         'January 1, 2000, 00:00 A.M.',
         'Month dd, YYYY, HH12:MI A.M.',
         'NLS_DATE_LANGUAGE = American'
       )
FROM   DUAL;
```

TO_DATE() nlsparam .

TO_DATE() :

```
SELECT TO_DATE( '2000-01-01 12:00:00', 'YYYY-MM-DD HH24:MI:SS' ) FROM DUAL;
```

TIMESTAMP .

```
CREATE TABLE date_table(
  date_value DATE
);

INSERT INTO date_table ( date_value ) VALUES ( TIMESTAMP '2000-01-01 12:00:00' );
```

TIMESTAMP DATE DATE . DATE CAST() .

```
SELECT CAST( TIMESTAMP '2000-01-01 12:00:00' AS DATE ) FROM DUAL;
```

Oracle DATE . (SQL / Plus, SQL / Developer, Toad, Java, Python) DATE DATE 7 8 .

DATE (SYSDATE DUMP() " 13") 8 (2012-11-26 16:41:09 2012-11-26 16:41:09) :

```
BYTE VALUE                          EXAMPLE
---- ------------------------------ --------------------------------------
1    Year modulo 256                220
2    Year multiples of 256          7   (7 * 256 + 220 = 2012)
3    Month                          11
4    Day                            26
5    Hours                          16
6    Minutes                        41
7    Seconds                        9
8    Unused                         0
```

DATE (DUMP() " 12") 7 (2012-11-26 16:41:09 .) :

```
BYTE VALUE                          EXAMPLE
---- ------------------------------ --------------------------------------
1    ( Year multiples of 100 ) + 100 120
2    ( Year modulo 100 ) + 100      112 ((120-100)*100 + (112-100) = 2012)
3    Month                          11
4    Day                            26
5    Hours + 1                      17
6    Minutes + 1                    42
7    Seconds + 1                    10
```

(,) . SQL TO_CHAR(date, format_model, nls_params) .

TO_CHAR(date [, format_model [, nls_params]]) :

(: *NLS_DATE_FORMAT* . .

```
CREATE TABLE table_name (
  date_value DATE
);

INSERT INTO table_name ( date_value ) VALUES ( DATE '2000-01-01' );
INSERT INTO table_name ( date_value ) VALUES ( TIMESTAMP '2016-07-21 08:00:00' );
INSERT INTO table_name ( date_value ) VALUES ( SYSDATE );
```

:

```
SELECT TO_CHAR( date_value, 'YYYY-MM-DD' ) AS formatted_date FROM table_name;
```

:

```
FORMATTED_DATE
--------------
2000-01-01
2016-07-21
2016-07-21
```

:

```
SELECT TO_CHAR(
         date_value,
         'FMMonth d yyyy, hh12:mi:ss AM',
```

```
         'NLS_DATE_LANGUAGE = French'
       ) AS formatted_date
FROM   table_name;
```

:

```
FORMATTED_DATE
------------------------------
Janvier   01 2000, 12:00:00 AM
Juillet   21 2016, 08:00:00 AM
Juillet   21 2016, 19:08:31 PM
```

Oracle DATE (TO_CHAR() TO_DATE()) NLS_DATE_FORMAT . .

.

```
SELECT VALUE FROM NLS_SESSION_PARAMETERS WHERE PARAMETER = 'NLS_DATE_FORMAT';
```

.

```
ALTER SESSION SET NLS_DATE_FORMAT = 'YYYY-MM-DD HH24:MI:SS';
```

(: .)

NLS_DATE_FORMAT TO_DATE() TO_CHAR() .

SQL / Plus SQL Developer

SQL / Plus SQL Developer ( ).

NLS_DATE_FORMAT .

- ,, /

DATE (/) .

```
SELECT DATE '2016-03-23' - DATE '2015-12-25' AS difference FROM DUAL;
```

.

```
DIFFERENCE
----------
        89
```

:

```
SELECT TO_DATE( '2016-01-02 01:01:12', 'YYYY-MM-DD HH24:MI:SS' )
         - TO_DATE( '2016-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS' )
         AS difference
FROM   DUAL
```

.

```
DIFFERENCE
----------
    1.0425
```

, 24 , 24*60 24*60*60 .

,, .

```
SELECT TRUNC( difference                       ) AS days,
       TRUNC( MOD( difference * 24,       24 ) ) AS hours,
       TRUNC( MOD( difference * 24*60,    60 ) ) AS minutes,
       TRUNC( MOD( difference * 24*60*60, 60 ) ) AS seconds
FROM   (
  SELECT TO_DATE( '2016-01-02 01:01:12', 'YYYY-MM-DD HH24:MI:SS' )
         - TO_DATE( '2016-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS' )
         AS difference
  FROM   DUAL
);
```

(: *FLOOR()* *TRUNC()* .

:

```
DAYS HOURS MINUTES SECONDS
---- ----- ------- -------
   1     1       1      12
```

NUMTODSINTERVAL() .

```
SELECT EXTRACT( DAY    FROM difference ) AS days,
       EXTRACT( HOUR   FROM difference ) AS hours,
       EXTRACT( MINUTE FROM difference ) AS minutes,
       EXTRACT( SECOND FROM difference ) AS seconds
FROM   (
  SELECT NUMTODSINTERVAL(
           TO_DATE( '2016-01-02 01:01:12', 'YYYY-MM-DD HH24:MI:SS' )
             - TO_DATE( '2016-01-01 00:00:00', 'YYYY-MM-DD HH24:MI:SS' ),
           'DAY'
         ) AS difference
  FROM   DUAL
);
```

-

MONTHS_BETWEEN(date1, date2) MONTHS_BETWEEN(date1, date2) .

```
SELECT MONTHS_BETWEEN( DATE '2016-03-10', DATE '2015-03-10' ) AS difference FROM DUAL;
```

:

```
DIFFERENCE
----------
        12
```

31 .

```
SELECT MONTHS_BETWEEN( DATE '2015-02-15', DATE '2015-01-01' ) AS difference FROM DUAL;
```

:

```
DIFFERENCE
----------
 1.4516129
```

MONTHS_BETWEEN 31 MONTHS_BETWEEN .

:

```
SELECT MONTHS_BETWEEN( DATE'2016-02-01', DATE'2016-02-01' - INTERVAL '1' DAY ) AS "JAN-FEB",
       MONTHS_BETWEEN( DATE'2016-03-01', DATE'2016-03-01' - INTERVAL '1' DAY ) AS "FEB-MAR",
       MONTHS_BETWEEN( DATE'2016-04-01', DATE'2016-04-01' - INTERVAL '1' DAY ) AS "MAR-APR",
       MONTHS_BETWEEN( DATE'2016-05-01', DATE'2016-05-01' - INTERVAL '1' DAY ) AS "APR-MAY"
FROM   DUAL;
```

:

```
JAN-FEB FEB-MAR MAR-APR APR-MAY
------- ------- ------- -------
0.03226 0.09677 0.03226 0.06452
```

12 .

,,,,

DATE , EXTRACT([YEAR | MONTH | DAY] FROM datevalue) .

```
SELECT EXTRACT (YEAR  FROM DATE '2016-07-25') AS YEAR,
       EXTRACT (MONTH FROM DATE '2016-07-25') AS MONTH,
       EXTRACT (DAY   FROM DATE '2016-07-25') AS DAY
FROM DUAL;
```

:

```
YEAR MONTH DAY
---- ----- ---
2016     7  25
```

(,) .

- CAST(datevalue AS TIMESTAMP) DATE TIMESTAMP EXTRACT([HOUR | MINUTE | SECOND] FROM

timestampvalue) ;.

- TO_CHAR(datevalue, format_model) .

:

```
SELECT EXTRACT( HOUR   FROM CAST( datetime AS TIMESTAMP ) ) AS Hours,
       EXTRACT( MINUTE FROM CAST( datetime AS TIMESTAMP ) ) AS Minutes,
       EXTRACT( SECOND FROM CAST( datetime AS TIMESTAMP ) ) AS Seconds
FROM   (
  SELECT TO_DATE( '2016-01-01 09:42:01', 'YYYY-MM-DD HH24:MI:SS' ) AS datetime FROM DUAL
);
```

:

```
HOURS MINUTES SECONDS
----- ------- -------
    9      42       1
```

DATE .

:

- TIMESTAMP WITH TIME ZONE .
- .

DATE UTC (Coordinated Universal Time) .

```
SELECT FROM_TZ(
         CAST(
           TO_DATE( '2016-01-01 12:00:00', 'YYYY-MM-DD HH24:MI:SS' )
           AS TIMESTAMP
         ),
         'UTC'
       )
       AT LOCAL AS time
FROM   DUAL;
```

ALTER SESSION SET TIME_ZONE = '+01:00'; ALTER SESSION SET TIME_ZONE = '+01:00'; .

```
TIME
-----------------------------------
2016-01-01 13:00:00.000000000 +01:00
```

ALTER SESSION SET TIME_ZONE = 'PST'; .

```
TIME
-----------------------------------
2016-01-01 04:00:00.000000000 PST
```

Oracle . My Oracle Support note 2019397.2 730795.1 .

TO_CHAR(date_value, 'D') _ TO_CHAR(date_value, 'D') .

NLS_TERRITORY .

```
ALTER SESSION SET NLS_TERRITORY = 'AMERICA';        -- First day of week is Sunday
SELECT TO_CHAR( DATE '1970-01-01', 'D' ) FROM DUAL;
```

5

```
ALTER SESSION SET NLS_TERRITORY = 'UNITED KINGDOM'; -- First day of week is Monday
SELECT TO_CHAR( DATE '1970-01-01', 'D' ) FROM DUAL;
```

4

NLS ( ) iso-week ( ) .

```
SELECT TRUNC( date_value ) - TRUNC( date_value, 'IW' ) + 1 FROM DUAL
```

: https://riptutorial.com/ko/oracle/topic/2087/

9:

Examples

( ) `DATE` ( ) `TIMESTAMP` ( ) . :

:

```
select to_char(sysdate + 1, 'YYYY-MM-DD') as tomorrow from dual;
```

:

```
select to_char(sysdate - 1, 'YYYY-MM-DD') as yesterday from dual;
```

5 .

```
select to_char(sysdate + 5, 'YYYY-MM-DD') as five_days_from_now from dual;
```

5 .

```
select to_char(sysdate + (5/24), 'YYYY-MM-DD HH24:MI:SS') as five_hours_from_now from dual;
```

10 .

```
select to_char(sysdate + (10/1440), 'YYYY-MM-DD HH24:MI:SS') as ten_mintues_from_now from
dual;
```

7 .

```
select to_char(sysdate + (7/86400), 'YYYY-MM-DD HH24:MI:SS') as seven_seconds_from_now from
dual;
```

`hire_date` 30 .

```
select * from emp where hire_date < sysdate - 30;
```

1 `last_updated` .

```
select * from logfile where last_updated >= sysdate - (1/24);
```

(: 1.5 , 36 , 2) `INTERVAL` . `DATE` W `TIMESTAMP` 2 . :

```
select * from logfile where last_updated >= sysdate - interval '1' hour;
```

Add_months

: `add_months(p_date, integer) return date;`

Add_months p_date amt .

```
SELECT add_months(date'2015-01-12', 2) m FROM dual;
```

[redacted]
2015-03-12

amt

```
SELECT add_months(date'2015-01-12', -2) m FROM dual;
```

[redacted]
2014-11-12

.

```
SELECT to_char( add_months(date'2015-01-31', 1),'YYYY-MM-DD') m FROM dual;
```

[redacted]
2015-02-28

: https://riptutorial.com/ko/oracle/topic/768/-

10:

.

USER_ , ALL_ DBA_ (ALL_) SYSDBA (DBA_) (USER_). , ALL_TABLES .

V$.

_PRIVS , .

: /

Examples

USER_SOURCE . OWNER .

```
select * from user_source where type='TRIGGER' and lower(text) like '%order%'
```

ALL_SOURCE .

```
select * from all_source where owner=:owner
```

DBA_SOURCE .

```
select * from dba_source
```

```
select owner, table_name
from all_tables
```

ALL_TAB_COLUMNS , . COLS USER_TAB_COLUMNS .

```
select *
from all_tab_columns
where table_name = :tname
```

.

```
select *
from dba_role_privs
where grantee= :username
```

:

1.

```
select *
from dba_sys_privs
```

```
where grantee = :username
```

2.

```
select *
from dba_tab_privs
where grantee = :username
```

.

.

```
select *
from role_role_privs
where role in (select granted_role from dba_role_privs where grantee= :username)
```

1.

```
select *
from role_sys_privs
where role in (select granted_role from dba_role_privs where grantee= :username)
```

2.

```
select *
from role_tab_privs
where role in (select granted_role from dba_role_privs where grantee= :username)
```

```
select *
from v$version
```

.

```
select *
from dba_objects
```

```
select * from dict
```

: https://riptutorial.com/ko/oracle/topic/7347/-

11:

/ .

Examples

Datapump

.

1. :

```
   select * from dba_datapump_jobs;
   SELECT * FROM DBA_DATAPUMP_SESSIONS;
   select username,opname,target_desc,sofar,totalwork,message from V$SESSION_LONGOPS where
username = 'bkpadmin';
```

2. :

- /
- :
- /

```
impdp <bkpadmin>/<bkp123> attach=<SYS_IMPORT_SCHEMA_01>
Import> status
```

/ **CTRL + C** .

3/6 :

```
create or replace directory DATAPUMP_REMOTE_DIR as '/oracle/scripts/expimp';
```

7 :

:

```
expdp <bkpadmin>/<bkp123>  parfile=<exp.par>
```

* <> . / . . *

- :
-
- [exp.par] :

```
schemas=<schema>
directory= DATAPUMP_REMOTE_DIR
dumpfile=<dbname>_<schema>.dmp
```

```
logfile=exp_<dbname>_<schema>.log
```

- :
- : . *(: CPU)*
- [exp.par] :

```
schemas=<schema>
directory= DATAPUMP_REMOTE_DIR
dumpfile=<dbname>_<schema>_%U.dmp
logfile=exp_<dbname>_<schema>.log
compression = all
parallel=5
```

- : [ ]
- [exp.par] :

```
tables= tname1, tname2, tname3
directory= DATAPUMP_REMOTE_DIR
dumpfile=<dbname>_<schema>.dmp
logfile=exp_<dbname>_<schema>.log
```

---

9 :

:

- .

:

```
impdp <bkpadmin>/<bkp123>  parfile=<imp.par>
```

* <> . / . . *

- :
-
- [say imp.par] :

```
schemas=<schema>
directory= DATAPUMP_REMOTE_DIR
dumpfile=<dbname>_<schema>.dmp
logfile=imp_<dbname>_<schema>.log
```

- :
- : *(: CPU)*
- [say imp.par] :

```
schemas=<schema>
```

```
directory= DATAPUMP_REMOTE_DIR
dumpfile=<dbname>_<schema>_%U.dmp
logfile=imp_<dbname>_<schema>.log
parallel=5
```

- : [ ]
- [say imp.par] :

```
tables= tname1, tname2, tname3
directory= DATAPUMP_REMOTE_DIR
dumpfile=<dbname>_<schema>.dmp
logfile=exp_<dbname>_<schema>.log
TABLE_EXISTS_ACTION= <APPEND /SKIP /TRUNCATE /REPLACE>
```

1.

[ ]	[ ]
1. .	4. .
2. .	5. .
3. 1 .	6. 4 .
7. .	
8. / SCP.	
	9. Import
	10. ,

```
 expdp <bkpadmin>/<bkp123> directory=DATAPUMP_REMOTE_DIR dumpfile=<customer.dmp>


 impdp <bkpadmin>/<bkp123> directory=DATAPUMP_REMOTE_DIR dumpfile=<customer.dmp>
remap_schema=<source schema>:<target schema> remap_tablespace=<source tablespace>:<target
tablespace>
```

: https://riptutorial.com/ko/oracle/topic/9391/-

12:

Examples

```
CREATE DATABASE LINK dblink_name
CONNECT TO remote_username
IDENTIFIED BY remote_password
USING 'tns_service_name';
```

DB .

```
SELECT * FROM MY_TABLE@dblink_name;
```

.

```
SELECT * FROM DUAL@dblink_name;
```

`USING` . :

```
USING 'tns_service_name.WORLD'
```

Oracle .

Oracle :

- 10g : https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_5005.htm
- 11g : https://docs.oracle.com/cd/B28359_01/server.111/b28310/ds_concepts002.htm
- 12g : https://docs.oracle.com/database/121/SQLRF/statements_5006.htm#SQLRF01205

"ORA1" "ORA2" . "ORA1" "ORA2" .

: `CREATE DATABASE LINK .` `CREATE PUBLIC DATABASE LINK .`

* Oracle Net .

:

ORA1 :

```
SQL> create <public> database link ora2 connect to user1 identified by pass1 using <tns name
of ora2>;
```

.

DB ORA1 .

```
SQL> Select name from V$DATABASE@ORA2; -- should return ORA2
```

`user1` ORA2 `SELECT` (TABLE1), "ORA1" "ORA2" DB :

```
SELECT COUNT(*) FROM TABLE1@ORA2;
```

requistes :

- ( ).
- .
- TNS .
- user1 ORA2 .
- user1 `SELECT` ORA2 .

: https://riptutorial.com/ko/oracle/topic/3859/-

13: SQL

SQL SQL . . . PL / SQL . PL / SQL DDL SQL .

:

1. . .

```
execute immediate 'select value from my_table where id = ' ||
     id_valiable into result_variable;
```

.

```
execute immediate 'select value from my_table where id = :P '
    using id_valiable into result_variable;
```

. . SQL . `1 or 1 = 1` `UPDATE` .

```
execute immediate 'update my_table set value = ''I have bad news for you'' where id = '
|| id;
```

. .

2. DDL .

Examples

SQL

. .

```
 function get_value(p_table_name varchar2, p_id number) return varchar2 is
    value varchar2(100);
  begin
    execute immediate 'select column_value from ' || p_table_name ||
                      ' where id = :P' into value using p_id;
    return value;
  end;
```

.

```
declare
  table_name varchar2(30) := 'my_table';
  id number := 1;
begin
  dbms_output.put_line(get_value(table_name, id));
end;
```

:

```
create table my_table (id number, column_value varchar2(100));
insert into my_table values (1, 'Hello, world!');
```

SQL

.

```
declare
  query_text varchar2(1000) := 'insert into my_table(id, column_value) values (:P_ID,
:P_VAL)';
  id number := 2;
  value varchar2(100) := 'Bonjour!';
begin
  execute immediate query_text using id, value;
end;
/
```

SQL

.

```
declare
  query_text varchar2(1000) := 'update my_table set column_value = :P_VAL where id = :P_ID';
  id number := 2;
  value varchar2(100) := 'Bonjour le monde!';
begin
  execute immediate query_text using value, id;
end;
/
```

DDL

.

```
begin
  execute immediate 'create table my_table (id number, column_value varchar2(100))';
end;
/
```

. SQL .

```
declare
  query_text varchar2(1000) := 'begin :P_OUT := cos(:P_IN); end;';
  in_value number := 0;
  out_value number;
begin
  execute immediate query_text using out out_value, in in_value;
  dbms_output.put_line('Result of anonymous block: ' || to_char(out_value));
end;
/
```

SQL : https://riptutorial.com/ko/oracle/topic/10905/-sql

14:

DUAL VARCHAR2(1) DUMMY x .

DUAL SYS . .

DUAL .

DUAL SQL . .

Examples

.

```
select sysdate from dual
```

start_value end_value .

```
select :start_value + level -1 n
from dual
connect by level <= :end_value  - :start_value + 1
```

: https://riptutorial.com/ko/oracle/topic/7328/-

15:

level N .

Examples

N

```
SELECT ROWNUM NO FROM DUAL CONNECT BY LEVEL <= 10
```

/ * . 1..100 * /

```
select level from dual connect by level <= 100;
```

/ * . 10 * /

```
select to_date('01-01-2017','mm-dd-yyyy')+level-1 as  dates from dual connect by level <= 10;
```

01-1 -17
02-JAN-17
03-1-2004
04-JAN-17
05-JAN-17
06-JAN-17
07-JAN-17
08-JAN-17
09-JAN-17
10-JAN-17

: https://riptutorial.com/ko/oracle/topic/6548/-

16:

- UPDATE [[AS]] SET = [, = }] * [WHERE]
- UPDATE - SET - = [, =] *

Examples

```
UPDATE
      TESTTABLE
SET
      TEST_COLUMN= 'Testvalue',TEST_COLUMN2= 123
WHERE
      EXISTS
          (SELECT MASTERTABLE.TESTTABLE_ID
           FROM MASTERTABLE
           WHERE ID_NUMBER=11);
```

(Oracle)

:

```
UPDATE
      (SELECT
            TESTTABLE.TEST_COLUMN AS OLD,
            'Testvalue' AS NEW
       FROM
                TESTTABLE
            INNER JOIN
                MASTERTABLE
            ON TESTTABLE.TESTTABLE_ID = MASTERTABLE.TESTTABLE_ID
       WHERE ID_NUMBER=11) T
SET
      T.OLD        = T.NEW;
```

```
MERGE INTO
      TESTTABLE
USING
      (SELECT
            T1.ROWID AS RID,
            T2.TESTTABLE_ID
       FROM
                TESTTABLE T1
            INNER JOIN
                MASTERTABLE T2
            ON TESTTABLE.TESTTABLE_ID = MASTERTABLE.TESTTABLE_ID
       WHERE ID_NUMBER=11)
ON
      ( ROWID = RID )
WHEN MATCHED
THEN
    UPDATE SET TEST_COLUMN= 'Testvalue';
```

```
drop table table01;
```

```
drop table table02;

create table table01 (
       code int,
       name varchar(50),
       old int
);

create table table02 (
       code int,
       name varchar(50),
       old int
);

truncate table table01;
insert into table01 values (1, 'A', 10);
insert into table01 values (9, 'B', 12);
insert into table01 values (3, 'C', 14);
insert into table01 values (4, 'D', 16);
insert into table01 values (5, 'E', 18);

truncate table table02;
insert into table02 values (1, 'AA', null);
insert into table02 values (2, 'BB', 123);
insert into table02 values (3, 'CC', null);
insert into table02 values (4, 'DD', null);
insert into table02 values (5, 'EE', null);

select * from table01 a order by 2;
select * from table02 a order by 2;

--

merge into table02 a using (
      select b.code, b.old from table01 b
) c on (
  a.code = c.code
)
when matched then update set a.old = c.old
;

--

select a.*, b.* from table01 a
inner join table02 b on a.code = b.codetable01;

select * from table01 a
where
       exists
       (
         select 'x' from table02 b where a.code = b.codetable01
       );

select * from table01 a where a.code in (select b.codetable01 from table02 b);

--

select * from table01 a
where
       not exists
       (
```

```
          select 'x' from table02 b where a.code = b.codetable01
        );

select * from table01 a where a.code not in (select b.codetable01 from table02 b);
```

: https://riptutorial.com/ko/oracle/topic/4193/---

17:

Examples

: || concat ()

Oracle SQL PL / SQL `||` .

:

`customers` .

```
 id  firstname    lastname
---  -----------  ----------
  1  Thomas       Woody
```

:

```
 SELECT firstname || ' ' || lastname || ' is in my database.' as "My Sentence"
   FROM customers;
```

:

```
My Sentence
---------------------------------
Thomas Woody is in my database.
```

Oracle SQL `CONCAT(str1, str2)` .

:

:

```
 SELECT CONCAT(firstname, ' is in my database.') from customers;
```

:

```
Expr1
---------------------------------
Thomas is in my database.
```

UPPER .

```
SELECT UPPER('My text 123!') AS result FROM dual;
```

:

```
RESULT
------------
MY TEXT 123!
```

INITCAP

`INITCAP` / .

```
SELECT INITCAP('HELLO mr macdonald!') AS NEW FROM dual;
```

```
NEW
-------------------
Hello Mr Macdonald!
```

LOWER .

```
SELECT LOWER('HELLO World123!') text FROM dual;
```

:

```
121!
```

2 : `(\d\d)`

```
SELECT REGEXP_REPLACE ('2, 5, and 10 are numbers in this example', '(\d\d)', '#')
FROM dual;
```

:

```
'2, 5, and # are numbers in this example'
```

`\1` , `\2` , `\3` .

```
 SELECT REGEXP_REPLACE ('swap around 10 in that one ', '(.*)(\d\d )(.*)', '\3\2\1\3')
 FROM dual;
```

SUBSTR

`SUBSTR` .

```
SELECT SUBSTR('abcdefg',2,3) FROM DUAL;
```

:

```
bcd
```

SUBSTR .

```
SELECT SUBSTR('abcdefg',-4,2) FROM DUAL;
```

:

```
de
```

: SUBSTR(mystring,-1,1)

LTRIM / RTRIM

LTRIM RTRIM . ( ).

,

```
select LTRIM('<===>HELLO<===>', '=<>')
      ,RTRIM('<===>HELLO<===>', '=<>')
from dual;
```

:

```
HELLO<===>
<===>HELLO
```

: https://riptutorial.com/ko/oracle/topic/1518/-

18:

, , DML .

Examples

B-

```
CREATE INDEX ord_customer_ix ON orders (customer_id);
```

oracle b-tree . . B- 2 . , index . . :

- , .
- **B-** .
- DML . .
- .

```
CREATE BITMAP INDEX
emp_bitmap_idx
ON index_demo (gender);
```

- **.**
- . , .
- 3 .
- 2 . . 2 .
- RAM . Row-ID Oracle Row-ID .

```
CREATE INDEX first_name_idx ON user_data (UPPER(first_name));


SELECT *
FROM   user_data
WHERE  UPPER(first_name) = 'JOHN2';
```

- .
- (where) .
- **Upper ()** . .

: https://riptutorial.com/ko/oracle/topic/9978/

19:

- SEQUENCE SCHEMA.SEQUENCE {INCREMENT BY INTEGER | INTEGER | MAXVALUE INTEGER | NOMAXVALUE INTEGER | MINVALUE INTEGER | | | NOCYCLE INTEGER | | NOCACHE | | NOODER}

~	
~	
nominvalue	.

Examples

:

CREATE SEQUENCE . .

. . . .

. . .

v , v CURRVAL v NEXTVAL SQL. .

CREATE SEQUENCE .

CREATE ANY SEQUENCE .

: oe customers_seq . customers ID .

```
CREATE SEQUENCE customers_seq
START WITH     1000
INCREMENT BY   1
NOCACHE
NOCYCLE;
```

customers_seq.nextval 1000 1001 . 1 .

: https://riptutorial.com/ko/oracle/topic/3709/

20:

Oracle Real Application Security Oracle 12c . , , , .

Examples

.

: SELECT , INSERT , UPDATE , DELETE . .

```
XS$PRIVILEGE(
    name=>'privilege_name'
    [, implied_priv_list=>XS$NAME_LIST('"SELECT"', '"INSERT"', '"UPDATE"', '"DELETE"')]
)

XS$PRIVILEGE_LIST(
    XS$PRIVILEGE(...),
    XS$PRIVILEGE(...),
    ...
);
```

:

:

```
BEGIN
    SYS.XS_PRINCIPAL.CREATE_USER('user_name');
END;
```

:

```
BEGIN
    SYS.XS_PRINCIPAL.CREATE_USER(name => 'user_name', schema => 'schema_name');
END;

BEGIN
    SYS.XS_PRINCIPAL.SET_PASSWORD('user_name', 'password');
END;
CREATE PROFILE prof LIMIT
    PASSWORD_REUSE_TIME 1/4440
    PASSWORD_REUSE_MAX 3
    PASSWORD_VERIFY_FUNCTION Verify_Pass;

BEGIN
    SYS.XS_PRINCIPAL.SET_PROFILE('user_name', 'prof');
END;

BEGIN
    SYS.XS_PRINCIPAL.GRANT_ROLES('user_name', 'XSONNCENT');
END;
```

(:)

```
BEGIN
    SYS.XS_PRINCIPAL.SET_VERIFIER('user_name', '6DFF060084ECE67F', XS_PRINCIPAL.XS_SHA512");
END;
```

:

:

```
DECLARE
    st_date TIMESTAMP WITH TIME ZONE;
    ed_date TIMESTAMP WITH TIME ZONE;
BEGIN
    st_date := SYSTIMESTAMP;
    ed_date := TO_TIMESTAMP_TZ('2013-06-18 11:00:00 -5:00','YYYY-MM-DD HH:MI:SS');
    SYS.XS_PRINCIPAL.CREATE_ROLE
        (name => 'app_regular_role',
        enabled => TRUE,
        start_date => st_date,
        end_date => ed_date);
END;
```

: (authenatication)

```
BEGIN
    SYS.XS_PRINCIPAL.CREATE_DYNAMIC_ROLE
        (name => 'app_dynamic_role',
        duration => 40,
        scope => XS_PRINCIPAL.SESSION_SCOPE);
END;
```

:

:

- XSPUBLIC
- XSBYPASS
- XSSESSIONADMIN
- XSNAMESPACEADMIN
- XSPROVISIONER
- XSCACHEADMIN
- XSDISPATCHER

: ( )

- DBMS_AUTH : ( )
- EXTERNAL_DBMS_AUTH : ( )
- DBMS_PASSWD : ( )
- MIDTIER_AUTH : ( )
- XSAUTHENTICATED : ( )
- XSSWITCH : ( )

: https://riptutorial.com/ko/oracle/topic/10864/---

21: MAF

Examples

Binding

```
        ValueExpression ve = AdfmfJavaUtilities.getValueExpression(<binding>, String.class);
       String <variable_name> = (String) ve.getValue(AdfmfJavaUtilities.getELContext());
```

"" EL .

"variable_name" .

```
        ValueExpression ve = AdfmfJavaUtilities.getValueExpression(<binding>, String.class);
        ve.setValue(AdfmfJavaUtilities.getELContext(), <value>);
```

"" EL .

"value" .

```
    AdfELContext adfELContext = AdfmfJavaUtilities.getAdfELContext();
    MethodExpression me;
    me = AdfmfJavaUtilities.getMethodExpression(<binding>, Object.class, new Class[] { });
    me.invoke(adfELContext, new Object[] { });
```

"" EL .

javaScript

```
AdfmfContainerUtilities.invokeContainerJavaScriptFunction(AdfmfJavaUtilities.getFeatureId(),
<function>, new Object[] {
                                                        });
```

"function" js .

MAF : https://riptutorial.com/ko/oracle/topic/6352/-maf

22:

Examples

EXAMPLE Oracle ERR $ _EXAMPLE .

```
EXECUTE DBMS_ERRLOG.CREATE_ERROR_LOG('EXAMPLE', NULL, NULL, NULL, TRUE);
```

SQL :

```
insert into EXAMPLE (COL1) values ('example')
LOG ERRORS INTO ERR$_EXAMPLE reject limit unlimited;
```

: https://riptutorial.com/ko/oracle/topic/3505/-

23: PL / SQL

.

Examples

```
DECLARE
    -- declare a variable
    message varchar2(20);
BEGIN
  -- assign value to variable
  message := 'HELLO WORLD';

  -- print message to screen
  DBMS_OUTPUT.PUT_LINE(message);
END;
/
```

PL / SQL : https://riptutorial.com/ko/oracle/topic/6451/-pl---sql-

24:

.

1. .
2. (COMMIT ROLLBACK) DML .

Examples

.

```
CREATE OR REPLACE PROCEDURE log_errors
(
  p_calling_program IN VARCHAR2,
  p_error_code IN INTEGER,
  p_error_description IN VARCHAR2
)
IS
  PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
  INSERT INTO error_log
  VALUES
  (
  p_calling_program,
  p_error_code,
  p_error_description,
  SYSDATE,
  USER
  );
  COMMIT;
END log_errors;
```

PLSQL log_errors .

```
BEGIN
   DELETE FROM dept WHERE deptno = 10;
EXCEPTION
   WHEN OTHERS THEN
      log_errors('Delete dept',sqlcode, sqlerrm);
      RAISE;
END;

SELECT * FROM error_log;

CALLING_PROGRAM    ERROR_CODE    ERROR_DESCRIPTION
ERROR_DATETIME         DB_USER
Delete dept        -2292         ORA-02292: integrity constraint violated - child record found
08/09/2016             APEX_PUBLIC_USER
```

: https://riptutorial.com/ko/oracle/topic/6103/-

25:

Examples

Oracle .

ID . scrpit . ID "PK_S".

```
begin
  for i in (select a.table_name, c.column_name
              from user_constraints a, user_cons_columns c
             where a.CONSTRAINT_TYPE = 'R'
               and a.R_CONSTRAINT_NAME = 'PK_S'
               and c.constraint_name = a.constraint_name) loop


         execute immediate 'update ' || i.table_name || ' set ' || i.column_name ||
                      '=to_number(''1000'' || ' ||  i.column_name || ') ';



  end loop;

end;
```

Oracle

T1 "pk_t1" .

```
Begin
    For I in (select table_name, constraint_name from user_constraint t where
r_constraint_name='pk_t1') loop

Execute immediate ' alter table ' || I.table_name || ' disable constraint ' ||
i.constraint_name;

   End loop;
End;
```

: https://riptutorial.com/ko/oracle/topic/6040/

26:

(SO) . ( ) . . MERGE .

Examples

:

```
create table tgt ( id, val ) as
  select 1, 'a' from dual union all
  select 2, 'b' from dual
;

Table TGT created.

create table src ( id, val ) as
  select 1, 'x' from dual union all
  select 2, 'y' from dual
;

Table SRC created.

update
  ( select t.val as t_val, s.val as s_val
    from   tgt t inner join src s on t.id = s.id
  )
set t_val = s_val
;

SQL Error: ORA-01779: cannot modify a column which maps to a non key-preserved table
01779. 00000 -  "cannot modify a column which maps to a non key-preserved table"
*Cause:    An attempt was made to insert or update columns of a join view which
           map to a non-key-preserved table.
*Action:   Modify the underlying base tables directly.
```

`src.id` `src.val` `1` . ( ). , `src.id` - . " " .

`src.id` , . ! ( ) . `src.id` PK `tgt.id` `tgt.id` PK FK join . .

```
alter table src add constraint src_uc unique (id);

Table SRC altered.

update
  ( select t.val as t_val, s.val as s_val
    from   tgt t inner join src s on t.id = s.id
  )
set t_val = s_val
;

2 rows updated.

select * from tgt;
```

```
ID  VAL
--  ---
 1  x
 2  y
```

MERGE ( ) MERGE "Oracle ." Oracle .

: https://riptutorial.com/ko/oracle/topic/8061/-

27:

- Ratio_To_Report (expr) OVER (query_partition_clause)

Examples

Ratio_To_Report

.

```
--Data
CREATE TABLE Employees (Name Varchar2(30), Salary Number(10));
INSERT INTO Employees Values ('Bob',2500);
INSERT INTO Employees Values ('Alice',3500);
INSERT INTO Employees Values ('Tom',2700);
INSERT INTO Employees Values ('Sue',2000);
--Query
SELECT Name, Salary, Ratio_To_Report(Salary) OVER () As Ratio
FROM Employees
ORDER BY Salary, Name, Ratio;
--Output
NAME                           SALARY      RATIO
------------------------------ ---------- ----------
Sue                                  2000 .186915888
Bob                                  2500  .23364486
Tom                                  2700 .252336449
Alice                                3500 .327102804
```

: https://riptutorial.com/ko/oracle/topic/6669/-

28:

- CREATE [REPLACE] CONTEXT USING [.] ;
- CREATE [OR REPLACE] CONTEXT USING [.] INITIALIZED EXTERNALLY;
- CREATE [OR REPLACE] CONTEXT USING [schema.] INITIALIZED GLOBALLY;
- CREATE [OR REPLACE] CONTEXT USING [schema.] ACCESSED GLOBALLY;

OR REPLACE	
	- SYS_CONTEXT .
	. : .
INITIALIZED	Oracle Database .
EXTERNALLY	OCI .
GLOBALLY	LDAP .
ACCESSED GLOBALLY	. ID .

Oracle (12cR1) : http://docs.oracle.com/database/121/SQLRF/statements_5003.htm

Examples

```
CREATE CONTEXT my_ctx USING my_pkg;
```

my_pkg (:

```
CREATE PACKAGE my_pkg AS
  PROCEDURE set_ctx;
END my_pkg;

CREATE PACKAGE BODY my_pkg AS
  PROCEDURE set_ctx IS
  BEGIN
    DBMS_SESSION.set_context('MY_CTX','THE KEY','Value');
    DBMS_SESSION.set_context('MY_CTX','ANOTHER','Bla');
  END set_ctx;
END my_pkg;
```

```
my_pkg.set_ctx;
```

.

```
SELECT SYS_CONTEXT('MY_CTX','THE KEY') FROM dual;
```

```
Value
```

: https://riptutorial.com/ko/oracle/topic/2088/-

29: ( )

Examples

N

`FETCH` Oracle 12c R1 .

```
SELECT   val
FROM     mytable
ORDER BY val DESC
FETCH FIRST 5 ROWS ONLY;
```

FETCH :

```
SELECT * FROM (
   SELECT   val
   FROM     mytable
   ORDER BY val DESC
) WHERE ROWNUM <= 5;
```

SQL

```
SELECT val
FROM   (SELECT val, rownum AS rnum
        FROM   (SELECT val
                FROM   rownum_order_test
                ORDER BY val)
        WHERE rownum <= :upper_limit)
WHERE  rnum >= :lower_limit ;
```

.

N

rownum .

```
select * from
(
  select val from  mytable
) where rownum<=5
```

order by .

:

```
select * from
(
    select val from  mytable order by val desc
```

```
) where rownum<=5
```

```
select * from
(
    select val from  mytable order by val
) where rownum<=5
```

N M (Oracle 12c)

row_number () .

```
with t as (
  select col1
  , col2
  , row_number() over (order by col1, col2) rn
  from table
)
select col1
, col2
from t
where rn between N and M; -- N and M are both inclusive
```

12c `OFFSET FETCH` .

Oracle 12g +

```
SELECT Id, Col1
FROM TableName
ORDER BY Id
OFFSET 20 ROWS FETCH NEXT 20 ROWS ONLY;
```

```
SELECT Id,
   Col1
 FROM (SELECT Id,
           Col1,
           row_number() over (order by Id) RowNumber
      FROM TableName)
WHERE RowNumber BETWEEN 21 AND 40
```

Oracle 12g +

```
SELECT Id, Col1
FROM TableName
ORDER BY Id
OFFSET 5 ROWS;
```

```
SELECT Id,
   Col1
 FROM (SELECT Id,
           Col1,
           row_number() over (order by Id) RowNumber
      FROM TableName)
WHERE RowNumber > 20
```

( ) : https://riptutorial.com/ko/oracle/topic/4300/-------

30:

Examples

.

firm's_address * .

* "firm's_address" .

table name . "table"

* . . .

* ;

: https://riptutorial.com/ko/oracle/topic/6553/----

31:

. . .

Enterprise Edition .

Examples

ID .

```
CREATE TABLE orders (
    order_nr NUMBER(15),
    user_id VARCHAR2(2),
    order_value NUMBER(15),
    store_id NUMBER(5)
)
PARTITION BY HASH(store_id) PARTITIONS 8;
```

2 .

.

```
CREATE TABLE orders (
    order_nr NUMBER(15),
    user_id VARCHAR2(2),
    order_value NUMBER(15),
    store_id NUMBER(5)
)
PARTITION BY RANGE(order_value) (
    PARTITION p1 VALUES LESS THAN(10),
    PARTITION p2 VALUES LESS THAN(40),
    PARTITION p3 VALUES LESS THAN(100),
    PARTITION p4 VALUES LESS THAN(MAXVALUE)
);
```

```
SELECT * FROM user_tab_partitions;
```

ID .

```
CREATE TABLE orders (
    order_nr NUMBER(15),
    user_id VARCHAR2(2),
    order_value NUMBER(15),
    store_id NUMBER(5)
)
PARTITION BY LIST(store_id) (
    PARTITION p1 VALUES (1,2,3),
    PARTITION p2 VALUES(4,5,6),
    PARTITION p3 VALUES(7,8,9),
    PARTITION p4 VALUES(10,11)
);
```

```
ALTER TABLE table_name DROP PARTITION partition_name;
```

```
SELECT * FROM orders PARTITION(partition_name);
```

```
ALTER TABLE table_name TRUNCATE PARTITION partition_name;
```

```
ALTER TABLE table_name RENAME PARTITION p3 TO p6;
```

```
ALTER TABLE table_name
MOVE PARTITION partition_name TABLESPACE tablespace_name;
```

```
ALTER TABLE table_name
ADD PARTITION new_partition VALUES LESS THAN(400);
```

.

```
ALTER TABLE table_name SPLIT PARTITION old_partition
    AT (new_high_bound) INTO (PARTITION new_partition TABLESPACE new_tablespace,
    PARTITION old_partition)
```

```
ALTER TABLE table_name
  MERGE PARTITIONS first_partition, second_partition
  INTO  PARTITION  splitted_partition TABLESPACE new_tablespace
```

. DDL ("insert ... select" "create table ... as select") "" (DML (/) .

:

1. ("B") ("A") :

"A" "OLD_VALUES" "B"

```
ALTER TABLE "A" EXCHANGE PARTITION "OLD_VALUES" WITH TABLE "B";
```

: "B"( ) "OLD_VALUES" "".

2. :

"A" "OLD_VALUES" "B"

```
ALTER TABLE "A" EXCHANGE PARTITION "OLD_VALUES" WITH TABLE "B";
```

: "OLD_VALUES"( ) "B" "".

: , .

. ---> " https://docs.oracle.com/cd/E11882_01/server.112/e25523/part_admin002.htm#i1107555 "(

" ")

: https://riptutorial.com/ko/oracle/topic/3955/-

32:

Examples

Oracle 10g MEDIAN .

```
SELECT MEDIAN(SAL)
FROM EMP
```

.

DATETIME .

MEDIAN . N Oracle RN = (1 + (0.5 * (N-1)) (RN) . CRN = CEILING (RN) FRN = FLOOR (RN) .

Oracle 9i 0.5 MEDIAN PERCENTILE_CONT

```
SELECT PERCENTILE_CONT(.5) WITHIN GROUP(order by SAL)
FROM EMP
```

. , () - .

.

```
SELECT name, salary, VARIANCE(salary) "Variance"
FROM employees
```

STDDEV expr . . STDDEV_SAMP (NULL) STDDEV 1 0 STDDEV_SAMP .

Oracle Database VARIANCE .

. .

DISTINCT analytic_clause query_partition_clause . order_by_clause windowing_clause .

hr.employees .

hr employees .

```
SELECT STDDEV(salary) "Deviation"
FROM employees;

Deviation
----------
3909.36575
```

hire_date hr.employees Department 80 .

```
SELECT last_name, salary,
STDDEV(salary) OVER (ORDER BY hire_date) "StdDev"
FROM employees
WHERE department_id = 30;

LAST_NAME                    SALARY     StdDev
------------------------- ---------- ----------
Raphaely                      11000          0
Khoo                           3100 5586.14357
Tobias                         2800  4650.0896
```

: https://riptutorial.com/ko/oracle/topic/2283/-

33:

(DOP)	/ . 2, 4, 8, 16 .
	.

Examples

.

```
SELECT /*+ PARALLEL(8) */ first_name, last_name FROM employee emp;
```

. .

```
SELECT /*+ PARALLEL(emp,8) */ first_name, last_name FROM employee emp;
```

```
SELECT /*+ PARALLEL(table_alias,Degree of Parallelism) */ FROM table_name table_alias;
```

100 . DOP 2 50 . DOP 4 25 .

. .

USE_NL

.

: `use_nl(AB)`

A B . . NL .

```
SELECT /*+use_nl(e d)*/ *
FROM Employees E
JOIN Departments D on E.DepartmentID = D.ID
```

" DIRECT PATH ".

`APPEND` . , . . .

- .
- Oracle (sqlldr).
-

.

```
INSERT /*+append*/ INTO Employees
SELECT *
```

```
FROM Employees;
```

USE_HASH

.

: `use_hash(TableA [TableB] ... [TableN])`

"HASH Oracle ( )) ."

.

: (HASH JOIN) .

:

```
SELECT /*+use_hash(e d)*/ *
FROM Employees E
JOIN Departments D on E.DepartmentID = D.ID
```

FULL Oracle .

```
create table fullTable(id) as select level from dual connect by level < 100000;
create index idx on fullTable(id);
```

.

```
select count(1) from fullTable f where id between 10 and 100;
-------------------------------------------------------------------------------
| Id  | Operation         | Name | Rows  | Bytes | Cost (%CPU)| Time     |
-------------------------------------------------------------------------------
|   0 | SELECT STATEMENT  |      |     1 |    13 |     3   (0)| 00:00:01 |
|   1 |  SORT AGGREGATE   |      |     1 |    13 |            |          |
|*  2 |   INDEX RANGE SCAN| IDX  |     2 |    26 |     3   (0)| 00:00:01 |
-------------------------------------------------------------------------------
```

.

```
select /*+ full(f) */ count(1) from fullTable f where id between 10 and 100;
-----------------------------------------------------------------------------------
| Id  | Operation          | Name      | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------
|   0 | SELECT STATEMENT   |           |     1 |    13 |    47   (3)| 00:00:01 |
|   1 |  SORT AGGREGATE    |           |     1 |    13 |            |          |
|*  2 |   TABLE ACCESS FULL| FULLTABLE |     2 |    26 |    47   (3)| 00:00:01 |
-----------------------------------------------------------------------------------
```

Oracle (*11g*) SQL SGA . . .

```
SELECT /*+ result_cache */ number FROM main_table;
```

-

```
Number
------
   1
   2
   3
   4
   5
   6
   7
   8
   9
   10

Elapsed: 00:00:02.20
```

.

-

```
Number
------
   1
   2
   3
   4
   5
   6
   7
   8
   9
   10

Elapsed: 00:00:00.10
```

2.20 0.10 .

/ / . .

: https://riptutorial.com/ko/oracle/topic/1490/

S. No		Contributors
1	Oracle Database	Community, J. Chomel, Jeffrey Kemp, Jon Ericson, Kevin Montrose, Mark Stewart, Sanjay Radadiya, Steven Feuerstein, tonirush
2	NULL	Dalex, JeromeFr
3	Oracle Advanced Queuing (AQ)	Jon Theriault
4	Oracle Database 12C	Muntasir, Vahid
5	WITH (AKA)	B Samedi, MT0
6		Aleksej, B Samedi, Bakhtiar Hasan, Daniel Langemann, Erkan Haspulat, Pranav Shah, Robin James, SriniV, Sumner Evans
7		Arkadiusz Łukasiewicz, MT0
8		carlosb, MT0, Roman, tonirush
9		David Aldridge, Florin Ghita, Jeffrey Kemp, Mark Stewart, tonirush, zygimantus
10		Mark Stewart, Pancho, Slava Babin
11		Vidya Thotangare
12		carlosb, Daniel Langemann, g00dy, kasi
13	SQL	Dmitry
14		Slava Babin
15		Sanjay Radadiya, TechEnthusiast
16		nimour pristou, Nogueira Jr, SriniV
17		carlosb, Eric B., Florin Ghita, Francesco Serra, J. Chomel, J.Hudler, Jeffrey Kemp, Mark Stewart, SriniV, Thunder, walen, zhliu03
18		smshafiqulislam
19		Pranav Shah, SriniV

20		Ben H
21	MAF	Anand Raj
22		zygimantus
23	PL / SQL	Jon Heller, Skynet, Zohar Elkayam
24		phonetic_man
25		SSD
26		mathguy
27		Leigh Riffel
28		Jeffrey Kemp
29	( )	Ahmed Mohamed, Martin Schapendonk, Matas Vaitkevicius, Sanjay Radadiya, tonirush, trincot
30		dev
31		BobC, carlosb, ivanzg, JeromeFr, Kamil Islamov, Stephen Leppik, tonirush
32		Evgeniy K., Matas Vaitkevicius, ppeterka, Pranav Shah
33		Aleksej, Florin Ghita, Jon Heller, Mark Stewart, Pirate X