How do I do top 1 in Oracle?

▲

210

How do I do the following?

```
select top 1 Fname from MyTbl
```

▼

In [Oracle 11g](#)?

★

33

| sql | oracle | oracle11g | greatest-n-per-group |

edited Aug 26 '16 at 10:03

a_horse_with_no_name
312k   46   475   578

asked Aug 10 '10 at 17:10

Gold
1

Have a look at [How does one select the TOP N rows from a table](#) – del.ave Aug 10 '10 at 17:12

---

possible duplicate of [How do I limit the number of rows returned by an Oracle query after ordering?](#) – Dave Jarvis Aug 20 '13 at 18:29

---

3   Can you tell us the order according to which you want 'top 1'? – Andrew Wolfe Nov 26 '14 at 3:34

---

1   First of all you should never ever rely on the DB engine to do that, ever. If you want to know things like that, put in a sequencer. When you do that it is guaranteed that they will be numbered in the order they were inserted. – FlyingGuy Feb 9

luke.com/sql/partial-results/top-n-queries – Ilia Maskov Oct 28
'15 at 13:10

9 Answers

▲

225

▼

✓

If you want just a first selected row, you can:

```
select fname from MyTbl where rownum = 1
```

You can also use analytic functions to order and take the
top x:

```
select max(fname) over (rank() order by some_factor) from
```

edited Jun 12 '16 at 14:36

Peter Mortensen
14k   19   87   114

answered Aug 10 '10 at 17:13

mcpeterson
3,308   4   16   22

---

8   Not sure what the second example is supposed to achieve...
     – Patrick Marchand Aug 10 '10 at 18:08

---

44   This is good if you only want 1 row and don't care which. If
     you want specific rows, like the most recent record, you need
     to do the sort in a subselect, like Vash's answer. Oracle
     assigns rownums before the sort. – Scott Bailey Aug 10 '10 at
     19:50

---

3   @Scott yup. that is correct. And Patrick, good point I think the
     syntax is incorrect on that. It really should be a keep over
     (dense_rank() last...) – mcpeterson Aug 11 '10 at 16:10

an order inner query (as per examples below). – JulesLt Aug 12 '10 at 12:36

2   The syntax in not correct in: select max(fname) over (rank() order by some_factor) from MyTbl – Stéphane Gerber Nov 6 '13 at 11:43 ✎

```
SELECT *
  FROM (SELECT * FROM MyTbl ORDER BY Fname )
 WHERE ROWNUM = 1;
```

145

answered Aug 10 '10 at 17:13

Damian Leszczyński - Vash

25.7k   7   46   88

5   This answer correctly gets the TOP row (orders the results before restricting on ROWNUM). – JulesLt Aug 12 '10 at 12:42

This answer is not an exact translation - the original query doesn't have an ORDER BY, nor does it return all columns in the table. – OMG Ponies Aug 12 '10 at 15:37

I stand corrected (see below). Will switch votes once time is up. – JulesLt Aug 12 '10 at 17:24

4   @OMGPonies yeah. but its probably what most people actually want who come to this page via googling their problem – NimChimpsky Nov 19 '15 at 11:12

3   This for sure must be the winning answer in this thread. I might add a note that for `top X` one can change it to `WHERE ROWNUM <= X` – SomethingSomething Feb 20 '16 at 23:36

following.

25

```
SELECT * FROM   MYTABLE
--ORDER BY COLUMNNAME -OPTIONAL
OFFSET 0 ROWS FETCH NEXT 1 ROWS ONLY
```

edited Jun 12 '16 at 14:38

Peter Mortensen
14k   19   87   114

answered Apr 27 '15 at 15:02

MSK
420   6   12

6   Interesting command, I'm using 12c here and the `OFFSET 0 ROWS` apparently is not necessary, you can use `FETCH NEXT 1 ROWS ONLY` or even `FETCH FIRST ROW ONLY` , the order by is important or it will be equivalent to just using a `WHERE rownum = 1` . I've even tried it in an OUTER APPLY instruction and it worked like Ms-SQL's TOP function there. – Rafael Merlin Nov 6 '15 at 19:01 ✎

You are right @RafaelMerlin. After your post I recognized that OFFSET 0 ROWS is not necessary. It would be useful when retrieving data between top X and top Y. – MSK Jan 6 '16 at 10:25

1   More examples : oracle-base.com/articles/12c/... – FixFaier Jan 5 '18 at 10:30

You could use `ROW_NUMBER()` with a `ORDER BY` clause in sub-query and use this column in replacement of `TOP N` . This can be explained step-by-step.

9

See the below table which have two columns `NAME` and

NAME	DT_CREATED
A	February, 01 2015 00:00:00
A	February, 02 2015 00:00:00
A	February, 03 2015 00:00:00
A	February, 04 2015 00:00:00
B	February, 03 2015 00:00:00
B	February, 04 2015 00:00:00
B	February, 05 2015 00:00:00
B	February, 06 2015 00:00:00

If you need to take only the first two dates irrespective of
 NAME , you could use the below query. The logic has been
written inside query

```
-- The number of records can be specified in WHERE clause
SELECT RNO,NAME,DT_CREATED
FROM
(
    -- Generates numbers in a column in sequence in the or
    SELECT ROW_NUMBER() OVER (ORDER BY DT_CREATED) AS RNO,
    NAME,DT_CREATED
    FROM DEMOTOP
)TAB
WHERE RNO<3;
```

RESULT

RNO	NAME	DT_CREATED
1	A	February, 01 2015 00:00:00
2	A	February, 02 2015 00:00:00

In some situations, we need to select TOP N results respective to each NAME . In such case we can use PARTITION BY with an ORDER BY clause in sub-query. Refer the below query.

```
-- The number of records can be specified in WHERE clause
SELECT RNO,NAME,DT_CREATED
FROM
(
    --Generates numbers in a column in sequence in the order
    SELECT ROW_NUMBER() OVER (PARTITION BY NAME ORDER BY DT
    NAME,DT_CREATED
    FROM DEMOTOP
)TAB
WHERE RNO<3;
```

RESULT

RNO	NAME	DT_CREATED
1	A	February, 01 2015 00:00:00
2	A	February, 02 2015 00:00:00
1	B	February, 03 2015 00:00:00
2	B	February, 04 2015 00:00:00

answered Nov 6 '15 at 4:59

[Sarath Avanavu](#)
11.5k 7 46 66

Using ROW_NUMBER()... is more correct solution than in topic answer. One problem with this solution (and with max(field) variant too) that you cannot do things like "select ... (select ROW_NUMBER() ...) **for update**;" – Alexo Po. Dec 9 '15 at 8:33 ✏️

And it's sometimes very important in PL/SQL (sorry, failed to edit previous comment in 5 minutes limit). – Alexo Po. Dec 9 '15 at 8:39

I think I do not understand you. **for update** clause can be used when ROWID is "easily" preserved by Oracle. So grouping (and grouping due to analytic clause usage) hides real ROWID and rows cannot be locked. And second, CTE (`with (select ... ) as` clause) does not change anything to this problem, CTE just aims in reading and supporting queries. Right? @Sarath Avanavu – Alexo Po. Dec 11 '15 at 8:45

Note on myself. The problem with ROWID actually happens specifically because of **where RNO<3** condition, in this case value of RNO is not connected with ROWID so that is why Oracle cannot lock rows. – Alexo Po. Dec 11 '15 at 8:54

You can do something like

7

```sql
SELECT *
    FROM (SELECT Fname FROM MyTbl ORDER BY Fname )
WHERE rownum = 1;
```

You could also use the analytic functions **RANK** and/or **DENSE_RANK**, but **ROWNUM** is probably the easiest.

edited Dec 10 '13 at 5:20

answered Dec 9 '13 at 12:51

Suman
54   3   16

1   can you please help with some example of rank etc. –
    breakfreehg Sep 15 '14 at 15:04

```
select * from (
    select FName from MyTbl
)
where rownum <= 1;
```

6

edited Feb 6 '18 at 10:07

ViRuSTriNiTy
3,791   1   22   43

answered Aug 10 '10 at 17:13

a'r
27.5k   5   54   60

Use:

5

```
SELECT x.*
  FROM (SELECT fname
          FROM MyTbl) x
 WHERE ROWNUM = 1
```

If using Oracle9i+, you could look at [using analytic functions like ROW_NUMBER() but they won't perform as well as ROWNUM](#).

answered Aug 10 '10 at 17:31

OMG Ponies
263k   62   447   469

---

1   Nice answer but contains a tiny typo. Where you say Oracle9i+ shouldn't that be 8i? [download-west.oracle.com/docs/cd/A87860_01/doc/server.817/…](#) – Ian Carpenter Aug 10 '10 at 18:32

@carpenteri: True, analytics were available in 8i - can't

Small comment - Vash's answer below includes an ORDER BY on the inner query which is critical if you want the TOP value of fname, rather than 'first' (which can be anything, most likely first row inserted). Might be worth an edit? – JulesLt Aug 12 '10 at 12:39

@JulesLt: The query provided by the OP doesn't include an ORDER BY, so this is answer represents and exact translation to Oracle syntax. – OMG Ponies Aug 12 '10 at 15:36

My misunderstanding of the SQL SERVER TOP syntax (erroneously presumed that it was similar to FIRST in RANK, not ROWNUM). Voted up. – JulesLt Aug 12 '10 at 17:23

To select the first row from a table and to select one row from a table are two different tasks and need a different query. There are many possible ways to do so. Four of them are:

3

First

```
select  max(Fname) from MyTbl;
```

Second

```
select  min(Fname) from MyTbl;
```

Third

```
select  Fname from MyTbl  where rownum = 1;
```

Fourth

```
select  max(Fname)  from MyTbl  where  rowid  (select  max(row
```

edited Jun 12 '16 at 14:37

Peter Mortensen
14k   19   87   114

answered Mar 12 '14 at 15:37

Vikas Hardia
1,899   2   26   44

I had the same issue, and I can fix this with this solution:

2

```sql
select a.*, rownum
from (select Fname from MyTbl order by Fname DESC) a
where
rownum = 1
```

You can order your result before to have the first value on top.

Good luck

edited Dec 21 '17 at 17:17

Caius Jard
13.3k   2   14   41

answered Jul 27 '13 at 23:12

user2607028
200   3   10

---

protected by cassiomolin Feb 22 at 13:50

association bonus does not count).

Would you like to answer one of these unanswered questions instead?