RowNum is a pseudo column that can be used to number and limit the rows as they are returned to SQL*Plus. You can also think of RowNum as a function that returns a row number value. The use of RowNum is limited to Oracle SQL.

The tests that work correctly are limited to testing if RowNum is **< value** or **<= value**. Think of RowNum as counting off the rows as they are passed into the result set until the row count reaches the desired number.

RowNum is applied to the rows **before** any Order By or Distinct. This means the results might not be what you had hoped to achieved.

Demo 01: These are a few of the rows from the products table sorted by list price with the most expensive first.

```
select prod_id, prod_name, prod_list_price
from prd_products
order by prod_list_price desc;
```

```
   PROD_ID PROD_NAME                 PROD_LIST_PRICE
---------- ------------------------- ---------------
      1126 WasherDryer                            850
      4568 Deluxe Cat Bed                      549.99
      4567 Deluxe Cat Tree                     549.99
      1120 Washer                              549.99
      1125 Dryer                                  500
      1040 Treadmill                           349.95
      4569 Mini Dryer                          349.95
      1050 Stationary bike                     269.95
      1060 Mountain bike                       255.95
      1010 Weights                                150
. . . rows omitted
```

Demo 02: Now we try to get the five most expensive products using RowNum

```
select prod_id, prod_name, prod_list_price
from prd_products
where ROWNUM <= 5
order by prod_list_price desc;
```

```
PROD_I PROD_NAME                PROD_LIST_PRICE
------ ------------------------ ---------------
1010   Weights                              150
1000   Hand Mixer                           125
1030   Basketball                         29.95
1070   Iron                                25.5
1020   Dartboard                          12.95

5 rows selected.
```

That is not what we wanted. What RowNum does is take the first 5 rows that come from the prd_products table and are passed to the result set and then it sorts those 5 rows. We will eventually solve the problem of the five most expensive items.

The next two pair of demos also point out what is happening. The first gets 10 rows from the products table using rownum <= 10. The second adds a Distinct and only 8 rows of the 10 rows are returned.

Demo 03: With rownum <= 10

```
select prod_list_price
from prd_products
where ROWNUM <= 10;
```

```
PROD_LIST_PRICE
---------------
            125
            150
          12.95
          29.95
           25.5
           25.5
           25.5
             25
         149.99
          49.99

10 rows selected.
```

Demo 04: Including a Distinct and only 8 rows are returned

10 rows were taken from the original result set and that temporary result set was then filtered by the Distinct keyword, leaving only 8 rows.

```
select DISTINCT prod_list_price
from prd_products
where ROWNUM <= 10;
```

```
PROD_LIST_PRICE
---------------
             25
         149.99
           25.5
          49.99
          12.95
            125
          29.95
            150

8 rows selected.
```

Demo 05: Including RowNum in the select may help you see what is happening. We get 5 rows coming from the table in whatever order the database engine delivers them. After the 5 rows are retrieved, the order by clause is applied- here the sort is by the price. We do get rownum values 1, 2, 3, 4, and 5.

```
select ROWNUM, prod_id, prod_name, prod_list_price
from prd_products
where rownum <= 5
order by prod_list_price desc;
```

```
    ROWNUM PROD_I PROD_NAME                 PROD_LIST_PRICE
---------- ------ ------------------------- ---------------
         2 1010   Weights                               150
         1 1000   Hand Mixer                            125
         4 1030   Basketball                          29.95
         5 1070   Iron                                 25.5
         3 1020   Dartboard                           12.95
```

Some other tests might help you see how this works. All of these return 0 rows.

Demo 06:

```
select rownum, prod_id, prod_name, prod_list_price
from prd_products
where rownum = 5;
```

Demo 07:

```
select rownum, prod_id, prod_name, prod_list_price
from prd_products
where rownum between 5 and 10;
```

Demo 08:

```
select rownum, prod_id, prod_name, prod_list_price
from prd_products
where rownum >= 5;
```

The tests that work as we probably want are limited to testing if RowNum is **equal to or less than a value**. RowNum is not the same as a Top or Limit keyword that you might be familiar with from another dbms.

One common use of RowNum is to limit the output to a few rows to see what type of data we have in a large table.