

Lab 2. Sorting Query Results

This lab focuses on customizing how your query results look. By understanding how you can control and modify your result sets, you can provide more readable and meaningful data.

2.1. Returning Query Results in a Specified Order

PROBLEM

You want to display the names, job, and salaries of employees in department 10 in order based on their salary (from lowest to highest). You want to return the following result set:

```
ENAME JOB SAL
-----
MILLER CLERK 1300
CLARK  MANAGER 2450
KING  PRESIDENT 5000
```

SOLUTION

Use the ORDER BY clause:

```
select ename,job,sal   from emp           where deptno = 10           order by sal asc
```

DISCUSSION

The ORDER BY clause allows you to order the rows of your result set. The solution sorts the rows based on SAL in ascending order. By default, ORDER BY will sort in ascending order, and the ASC clause is therefore optional. Alternatively, specify DESC to sort in descending order:

```
select ename,job,sal   from emp           where deptno = 10           order by sal desc
```

ENAME	JOB	SAL
KING	PRESIDENT	5000
CLARK	MANAGER	2450
MILLER	CLERK	1300

You need not specify the name of the column on which to sort. You can instead specify a number representing the column. The number starts at 1 and matches the items in the SELECT list from left to right. For example:

```
select ename,job,sal   from emp           where deptno = 10           order by 3 desc
```

ENAME	JOB	SAL
KING	PRESIDENT	5000
CLARK	MANAGER	2450
MILLER	CLERK	1300

The number 3 in this example's ORDER BY clause corresponds to the third column in the SELECT list, which is SAL.

2.2. Sorting by Multiple Fields

PROBLEM

You want to sort the rows from EMP first by DEPTNO ascending, then by salary descending. You want to return the following result set:

```
EMPNO DEPTNO SAL ENAME JOB
-----
7839 10 5000 KING  PRESIDENT
7782 10 2450 CLARK  MANAGER
7934 10 1300 MILLER CLERK
7788 20 3000 SCOTT  ANALYST
7902 20 3000 FORD   ANALYST
7566 20 2975 JONES  MANAGER
```

7876 20 1100 ADAMS CLERK
7369 20 800 SMITH CLERK
7698 30 2850 BLAKE MANAGER
7499 30 1600 ALLEN SALESMAN
7844 30 1500 TURNER SALESMAN
7521 30 1250 WARD SALESMAN
7654 30 1250 MARTIN SALESMAN
7900 30 950 JAMES CLERK

SOLUTION

List the different sort columns in the ORDER BY clause, separated by commas:

```
1 select empno,deptno,sal,ename,job
2 from emp
3 order by deptno, sal desc
```

DISCUSSION

The order of precedence in ORDER BY is from left to right. If you are ordering using the numeric position of a column in the SELECT list, then that number must not be greater than the number of items in the SELECT list. You are generally permitted to order by a column not in the SELECT list, but to do so you must explicitly name the column. However, if you are using GROUP BY or DISTINCT in your query, you cannot order by columns that are not in the SELECT list.

2.3. Sorting by Substrings

PROBLEM

You want to sort the results of a query by specific parts of a string. For example, you want to return employee names and jobs from table EMP and sort by the last two characters in the job field. The result set should look like the following:

```
ENAME JOB
-----
KING PRESIDENT
SMITH CLERK
ADAMS CLERK
JAMES CLERK
MILLER CLERK
JONES MANAGER
CLARK MANAGER
BLAKE MANAGER
ALLEN SALESMAN
MARTIN SALESMAN
WARD SALESMAN
TURNER SALESMAN
SCOTT ANALYST
FORD ANALYST
```

SOLUTION

DB2, MySQL, Oracle, and PostgreSQL

Use the SUBSTR function in the ORDER BY clause:

```
select ename,job          from emp          order by substr(job,length(job)-1)
```

SQL Server

Use the SUBSTRING function in the ORDER BY clause:

```
select ename,job          from emp          order by substring(job,len(job)-1,2)
```

DISCUSSION

Using your DBMS's substring function, you can easily sort by any part of a string. To sort by the last two characters of a string, find the end of the string (which is the length of the string) and subtract 2. The start position will be the second to last character in the string. You then take all characters after that start position. Because SQL Server requires a third parameter in SUBSTRING to specify the number of characters to take. In this example, any number greater than or equal to 2 will work.

2.4. Sorting Mixed Alphanumeric Data

PROBLEM

You have mixed alphanumeric data and want to sort by either the numeric or character portion of the data. Consider this view:

```
create view V as select ename||' '||deptno as data from emp
```

```
select * from V
```

DATA

```
-----  
SMITH 20  
ALLEN 30  
WARD 30  
JONES 20  
MARTIN 30  
BLAKE 30  
CLARK 10  
SCOTT 20  
KING 10  
TURNER 30  
ADAMS 20  
JAMES 30  
FORD 20  
MILLER 10
```

You want to sort the results by DEPTNO or ENAME. Sorting by DEPTNO produces the following result set:

DATA

```
-----  
CLARK 10  
KING 10  
MILLER 10  
SMITH 20  
ADAMS 20  
FORD 20  
SCOTT 20  
JONES 20  
ALLEN 30  
BLAKE 30  
MARTIN 30  
JAMES 30  
TURNER 30  
WARD 30
```

Sorting by ENAME produces the following result set:

DATA

```
-----  
ADAMS 20  
ALLEN 30  
BLAKE 30  
CLARK 10  
FORD 20  
JAMES 30  
JONES 20  
KING 10  
MARTIN 30  
MILLER 10  
SCOTT 20  
SMITH 20  
TURNER 30  
WARD 30
```

SOLUTION

Oracle and PostgreSQL
Use the functions REPLACE and TRANSLATE to modify the string for sorting:
ORDER BY DEPTNO

```
select data
  from V
order by replace(data,
replace(
translate(data,'0123456789','#####'),'#',''),'')

ORDER BY ENAME
```

```
select data
from emp
order by replace(
translate(data,'0123456789','#####'),'#','')
```

DB2
Implicit type conversion is more strict in DB2 than in Oracle or PostgreSQL, so you will need to cast DEPTNO to a CHAR for view V to be valid. Rather than recreate view V, this solution will simply use an inline view. The solution uses REPLACE and TRANSLATE in the same way as the Oracle and PostgreSQL solution, but the order of arguments for TRANSLATE is slightly different for DB2:

```
ORDER BY DEPTNO

select *
from (
select ename||' '||cast(deptno as char(2)) as data
from emp
) v
order by replace(data,
replace(
translate(data,'#####','0123456789'),'#',''),'')

/* ORDER BY ENAME */

select *
from (
select ename||' '||cast(deptno as char(2)) as data
from emp
) v
order by replace(
translate(data,'#####','0123456789'),'#','')
```

MySQL and SQL Server
The TRANSLATE function is not currently supported by these platforms, thus a solution for this problem will not be provided.

DISCUSSION
The TRANSLATE and REPLACE functions remove either the numbers or characters from each row, allowing you to easily sort by one or the other. The values passed to ORDER BY are shown in the following query results (using the Oracle solution as the example, as the same technique applies to all three vendors; only the order of parameters passed to TRANSLATE is what sets DB2 apart):

```
select data,      replace(data,      replace(      translate(data,'0123456789','#####'),'#',''),'') nums,      replace(      transl
```

DATA	NUMS	CHARS
SMITH 20	20	SMITH
ALLEN 30	30	ALLEN
WARD 30	30	WARD
JONES 20	20	JONES
MARTIN 30	30	MARTIN
BLAKE 30	30	BLAKE
CLARK 10	10	CLARK
SCOTT 20	20	SCOTT
KING 10	10	KING
TURNER 30	30	TURNER
ADAMS 20	20	ADAMS
JAMES 30	30	JAMES
FORD 20	20	FORD
MILLER 10	10	MILLER

2.5. Dealing with Nulls when Sorting

PROBLEM

You want to sort results from EMP by COMM, but the field is nullable. You need a way to specify whether nulls sort last:

ENAME SAL COMM

```
-----  
TURNER 1500 0  
ALLEN 1600 300  
WARD 1250 500  
MARTIN 1250 1400  
SMITH 800  
JONES 2975  
JAMES 950  
MILLER 1300  
FORD 3000  
ADAMS 1100  
BLAKE 2850  
CLARK 2450  
SCOTT 3000  
KING 5000
```

or whether they sort first:

ENAME SAL COMM

```
-----  
SMITH 800  
JONES 2975  
CLARK 2450  
BLAKE 2850  
SCOTT 3000  
KING 5000  
JAMES 950  
MILLER 1300  
FORD 3000  
ADAMS 1100  
MARTIN 1250 1400  
WARD 1250 500  
ALLEN 1600 300  
TURNER 1500 0
```

SOLUTION

Depending on how you want the data to look (and how your particular RDBMS sorts NULL values), you can sort the nullable column in ascending or descending order:

```
select ename,sal,comm from emp order by select ename,sal,comm from emp order by 3 desc
```

This solution puts you in a position such that if the nullable column contains non-NULL values, they will be sorted in ascending or descending order as well, according to what you ask for; this may or may not be what you have in mind. If instead you would like to sort NULL values differently than non-NULL values, for example, you want to sort non-NULL values in ascending or descending order and all NULL values last, you can use a CASE expression to conditionally sort the column.

DB2, MySQL, PostgreSQL, and SQL Server

Use a CASE expression to “flag” when a value is NULL. The idea is to have a flag with two values: one to represent NULLs, the other to represent non-NULLs. Once you have that, simply add this flag column to the ORDER BY clause. You’ll easily be able to control whether NULL values are sorted first or last without interfering with non-NULL values:

```
/* NON-NULL COMM SORTED ASCENDING, ALL NULLS LAST */
```

```
select ename,sal,comm      from (      select ename,sal,comm,      case when comm is null then 0 else 1 end as is_null      from emp
```

ENAME	SAL	COMM
TURNER	1500	0
ALLEN	1600	300
WARD	1250	500
MARTIN	1250	1400
SMITH	800	
JONES	2975	
JAMES	950	
MILLER	1300	
FORD	3000	
ADAMS	1100	
BLAKE	2850	
CLARK	2450	
SCOTT	3000	
KING	5000	

```
/* NON-NULL COMM SORTED DESCENDING, ALL NULLS LAST */
```

```
select ename,sal,comm
  from (
select ename,sal,comm,
       case when comm is null then 0 else 1 end as is_null
  from emp
 ) x
 order by is_null desc,comm desc
```

ENAME	SAL	COMM
MARTIN	1250	1400
WARD	1250	500
ALLEN	1600	300
TURNER	1500	0
SMITH	800	
JONES	2975	
JAMES	950	
MILLER	1300	
FORD	3000	
ADAMS	1100	
BLAKE	2850	
CLARK	2450	
SCOTT	3000	
KING	5000	

```
/* NON-NULL COMM SORTED ASCENDING, ALL NULLS FIRST */
```

```
select ename,sal,comm
  from (
select ename,sal,comm,
       case when comm is null then 0 else 1 end as is_null
  from emp
 ) x
 order by is_null,comm
```

ENAME	SAL	COMM
SMITH	800	
JONES	2975	
CLARK	2450	
BLAKE	2850	
SCOTT	3000	
KING	5000	
JAMES	950	
MILLER	1300	
FORD	3000	
ADAMS	1100	
TURNER	1500	0
ALLEN	1600	300
WARD	1250	500
MARTIN	1250	1400

```
/* NON-NULL COMM SORTED DESCENDING, ALL NULLS FIRST */
```

```

select ename,sal,comm
  from (
select ename,sal,comm,
       case when comm is null then 0 else 1 end as is_null
  from emp
       ) x
 order by is_null,comm desc

```

ENAME	SAL	COMM
-----	-----	-----
SMITH	800	
JONES	2975	
CLARK	2450	
BLAKE	2850	
SCOTT	3000	
KING	5000	
JAMES	950	
MILLER	1300	
FORD	3000	
ADAMS	1100	
MARTIN	1250	1400
WARD	1250	500
ALLEN	1600	300
TURNER	1500	0

Oracle

Users on Oracle8i Database and earlier can use the solution for the other platforms. Users on Oracle9i Database and later can use the NULLS FIRST and NULLS LAST extension to the ORDER BY clause to ensure NULLs are sorted first or last regardless of how non-NULL values are sorted:

```

/* NON-NULL COMM SORTED ASCENDING, ALL NULLS LAST */

```

```

select ename,sal,comm   from emp           order by comm nulls last

```

ENAME	SAL	COMM
-----	-----	-----
TURNER	1500	0
ALLEN	1600	300
WARD	1250	500
MARTIN	1250	1400
SMITH	800	
JONES	2975	
JAMES	950	
MILLER	1300	
FORD	3000	
ADAMS	1100	
BLAKE	2850	
CLARK	2450	
SCOTT	3000	
KING	5000	

```
/* NON-NULL COMM SORTED ASCENDING, ALL NULLS FIRST */
```

```
select ename,sal,comm
       from emp
       order by comm nulls first
```

ENAME	SAL	COMM
-----	-----	-----
SMITH	800	
JONES	2975	
CLARK	2450	
BLAKE	2850	
SCOTT	3000	
KING	5000	
JAMES	950	
MILLER	1300	
FORD	3000	
ADAMS	1100	
TURNER	1500	0
ALLEN	1600	300
WARD	1250	500
MARTIN	1250	1400

```
/* NON-NULL COMM SORTED DESCENDING, ALL NULLS FIRST */
```

```
select ename,sal,comm
       from emp
       order by comm desc nulls first
```

ENAME	SAL	COMM
-----	-----	-----
SMITH	800	
JONES	2975	
CLARK	2450	
BLAKE	2850	
SCOTT	3000	
KING	5000	
JAMES	950	
MILLER	1300	
FORD	3000	
ADAMS	1100	
MARTIN	1250	1400
WARD	1250	500
ALLEN	1600	300
TURNER	1500	0

DISCUSSION

Unless your RDBMS provides you with a way to easily sort NULL values first or last without modifying non-NULL values in the same column (such as Oracle does), you'll need an auxiliary column.

TIP

As of the time of this writing, DB2 users can use NULLS FIRST and NULLS LAST in the ORDER BY subclause of the OVER clause in window functions but not in the ORDER BY clause for the entire result set.

The purpose of this extra column (in the query only, not in the table) is to allow you to identify NULL values and sort them altogether, first or last. The

following query returns the result set for inline view X for the non-Oracle solution:

```
select  ename,sal,comm,          case when comm is null then 0 else 1 end as is_null      from emp
```

ENAME	SAL	COMM	IS_NULL
SMITH	800		0
ALLEN	1600	300	1
WARD	1250	500	1
JONES	2975		0
MARTIN	1250	1400	1
BLAKE	2850		0
CLARK	2450		0
SCOTT	3000		0
KING	5000		0
TURNER	1500	0	1
ADAMS	1100		0
JAMES	950		0
FORD	3000		0
MILLER	1300		0

By using the values returned by IS_NULL, you can easily sort NULLS first or last without interfering with the sorting of COMM.

2.6. Sorting on a Data Dependent Key

PROBLEM

You want to sort based on some conditional logic. For example: if JOB is “SALESMAN” you want to sort on COMM; otherwise, you want to sort by SAL.
You want to return the following result set:

```
ENAME SAL JOB COMM
-----
TURNER 1500 SALESMAN 0
ALLEN 1600 SALESMAN 300
WARD 1250 SALESMAN 500
SMITH 800 CLERK
JAMES 950 CLERK
ADAMS 1100 CLERK
MILLER 1300 CLERK
MARTIN 1250 SALESMAN 1400
CLARK 2450 MANAGER
BLAKE 2850 MANAGER
JONES 2975 MANAGER
SCOTT 3000 ANALYST
FORD 3000 ANALYST
KING 5000 PRESIDENT
```

SOLUTION

Use a CASE expression in the ORDER BY clause:

```
select  ename,sal,job,comm      from emp      order by case when job = 'SALESMAN' then comm else sal end
```

DISCUSSION

You can use the CASE expression to dynamically change how results are sorted. The values passed to the ORDER BY look as follows:

```
select  ename,sal,job,comm,          case when job = 'SALESMAN' then comm else sal end as ordered      from emp      order by 5
```

ENAME	SAL	JOB	COMM	ORDERED
TURNER	1500	SALESMAN	0	0
ALLEN	1600	SALESMAN	300	300
WARD1	250	SALESMAN	500	500
SMITH	800	CLERK		800
JAMES	950	CLERK		950
ADAMS	1100	CLERK		1100
MILLER	1300	CLERK		1300
MARTIN	1250	SALESMAN	1400	1400
CLARK2	450	MANAGER		2450
BLAKE2	850	MANAGER		2850
JONES2	975	MANAGER		2975
SCOTT	3000	ANALYST		3000
FORD	3000	ANALYST		3000
KING	5000	PRESIDENT		5000