

Table of Contents

1. Selecting columns.....	1
2. Selecting all columns.....	2
3. Column aliases.....	2
4. Sorting the output display.....	3

In this discussion we will examine a few features of the Select statement. These are:

- selecting individual columns
- selecting all columns
- using column aliases
- sorting the rows displayed

1. Selecting columns

The first few queries use only two clauses: the FROM clause to identify the table that supplies the data and the SELECT clause to identify the columns to be returned. For these queries, all rows from the table are returned. This set of demos uses the zoo_2016 table. Your data set might be different depending on the rows you inserted. You indicate which columns you want displayed and the order of the columns by listing the column names in the Select clause.

Demo 01: You can display the columns in any order. Note that row for the animal with no name displays a bank cell with this client.

```
Select  z_type, z_name
From zoo_2016;
```

Z_TYPE	Z_NAME
-----	-----
Giraffe	Sam
Armadillo	Abigail
Lion	Leon
Lion	Lenora
Giraffe	Sally Robinson
Zebra	Huey
Zebra	Dewey
Zebra	Louie
Horse	
Giraffe	Dewey
Giraffe	Arnold
Giraffe	
Giraffe	
Giraffe	Geoff
armadillo	Anders
armadillo	Anne
Lion	Leon
Lion	
Lion	
Lion	

Demo 02: Display dates and numeric values. The default display for dates uses a format with a two digit day, a three letter month and a 2 digit year.

```
Select
  z_dob
, z_cost
, z_name
From zoo_2016;
```

Z_DOB	Z_COST	Z_NAME
15-MAY-14	5000	Sam
15-JAN-13	490	Abigail
25-FEB-10	5000	Leon
25-MAR-14	5000	Lenora
15-MAY-14	5000.25	Sally Robinson
02-JUN-13	2500.25	Huey
02-JUN-14	2500.25	Dewey
02-JAN-13	2500.25	Louie
10-JAN-15	490	
06-JUN-13	3750	Dewey
15-MAY-14	5000	Arnold
15-MAY-13	5000	
15-MAY-02	5000	
15-MAY-02	5000	Geoff
15-JAN-13	490	Anders
15-JAN-13	490.01	Anne
25-FEB-13	1850	Leon
25-FEB-13	1850	
25-FEB-13	1850	
25-FEB-13	1850	

2. Selecting all columns

The symbol `*` is used to indicate that all columns should be returned. This is inefficient if you do not need to see all of the columns but is helpful for a quick look at a small table.

Using `Select *` can be a bad idea with embedded SQL if the table design is changed. Embedded SQL refers to SQL statement that might be included inside other units of code. You also have to consider that someone might reorder the column positions in the table and then your query produces a different result.

Demo 03: Display all columns, all rows.

```
Select *
From zoo_2016
;
```

Z_ID	Z_NAME	Z_TYPE	Z_COST	Z_DOB	Z_ACQUIRE
23	Sam	Giraffe	5000	15-MAY-14	15-MAY-14
25	Abigail	Armadillo	490	15-JAN-13	15-APR-13
56	Leon	Lion	5000	25-FEB-10	25-MAR-10
57	Lenora	Lion	5000	25-MAR-14	31-MAR-14
85	Sally Robinson	Giraffe	5000.25	15-MAY-14	15-MAY-14
43	Huey	Zebra	2500.25	02-JUN-13	02-JUN-14
44	Dewey	Zebra	2500.25	02-JUN-14	02-JUN-14
45	Louie	Zebra	2500.25	02-JAN-13	02-JAN-13
. . . rows omitted					

3. Column aliases

By default, the column headers are the attribute names displayed in uppercase letters. Column aliases can be used to supply different headers for the output display. Column aliases are limited to 30 characters.

Notice in the demos below that the column aliases are in upper case.

You can also use the `SQL*Plus` column command to affect the final display. We will not use this command during most of the semester as we are focusing on the SQL proper.

Demo 04: Display column headers other than the attribute names. The word AS is optional and may be omitted. Most people include the word AS for the column aliases.

```

Select
  z_id
, z_dob AS BirthDate
, z_cost AS Price
, z_name AS NAME
From zoo_2016
;

```

Z_ID	BIRTHDATE	PRICE	NAME
23	15-MAY-14	5000	Sam
25	15-JAN-13	490	Abigail
56	25-FEB-10	5000	Leon
57	25-MAR-14	5000	Lenora
85	15-MAY-14	5000.25	Sally Robinson
43	02-JUN-13	2500.25	Huey
44	02-JUN-14	2500.25	Dewey
45	02-JAN-13	2500.25	Louie
. . . rows omitted			

Demo 05: The use of double quotes for your aliases allows you to use spaces or special characters in the header and also preserves the case. Note that the default column width that SQL*Plus uses for a date column truncates our column alias.

```

Select
  z_id
, z_dob AS "Date of Birth"
, z_cost AS "Price $"
, z_name As "Name"
From zoo_2016
;

```

Z_ID	Date of B	Price \$	Name
23	15-MAY-14	5000	Sam
25	15-JAN-13	490	Abigail
56	25-FEB-10	5000	Leon
57	25-MAR-14	5000	Lenora
85	15-MAY-14	5000.25	Sally Robinson
43	02-JUN-13	2500.25	Huey
44	02-JUN-14	2500.25	Dewey
45	02-JAN-13	2500.25	Louie
. . rows omitted			

4. Sorting the output display

If you want to control the order in which the rows are displayed, you use an ORDER BY clause.

You can order by

- a column
- a column alias
- the numeric position of the column in the Select (not always a good idea)
- a calculated column expression (we will discuss this in the next unit)

If you have two columns with the same alias and try to sort by the alias, you will get an error message.

Demo 06: Controlling the order in which the rows are displayed. This is sorted by price with the lower values first; this is an ascending sort which is the default sort order.

```
Select
  z_id
, z_cost AS "Price"
, z_name As "Name"
From zoo_2016
ORDER BY z_cost;
```

Z_ID	BirthDate	Price	Name
25	15-JAN-13	490	Abigail
370	15-JAN-13	490	Anders
47	10-JAN-15	490	
371	15-JAN-13	490.01	Anne
374	25-FEB-13	1850	
375	25-FEB-13	1850	
373	25-FEB-13	1850	
372	25-FEB-13	1850	Leon
44	02-JUN-14	2500.25	Dewey
45	02-JAN-13	2500.25	Louie
43	02-JUN-13	2500.25	Huey
52	06-JUN-13	3750	Dewey
56	25-FEB-10	5000	Leon
23	15-MAY-14	5000	Sam
259	15-MAY-02	5000	
258	15-MAY-13	5000	
257	15-MAY-14	5000	Arnold
57	25-MAR-14	5000	Lenora
260	15-MAY-02	5000	Geoff
85	15-MAY-14	5000.25	Sally Robinson

Demo 07: Using DESC to specify a descending sort.

```
Select
  z_id
, z_cost AS "Price"
, z_name As "Name"
From zoo_2016
ORDER BY z_cost DESC;
```

Z_ID	BirthDate	Price	Name
85	15-MAY-14	5000.25	Sally Robinson
56	25-FEB-10	5000	Leon
57	25-MAR-14	5000	Lenora
259	15-MAY-02	5000	
258	15-MAY-13	5000	
23	15-MAY-14	5000	Sam
260	15-MAY-02	5000	Geoff
257	15-MAY-14	5000	Arnold
52	06-JUN-13	3750	Dewey
45	02-JAN-13	2500.25	Louie
44	02-JUN-14	2500.25	Dewey
43	02-JUN-13	2500.25	Huey
374	25-FEB-13	1850	
375	25-FEB-13	1850	
373	25-FEB-13	1850	
372	25-FEB-13	1850	Leon
371	15-JAN-13	490.01	Anne
370	15-JAN-13	490	Anders
25	15-JAN-13	490	Abigail
47	10-JAN-15	490	

If two rows have the same value for `z_cost`, then we have not specified an exact order for those rows

Demo 08: This is a two level sort. The first sort key is the `z_type`. If the `z_type` of two rows match, then the cost is used for the second sort level.
The other thing to note here is that the case of the `z_type` values is considered. Oracle is case sensitive on sorting character data and uppercase letters sort before lowercase. You can see that `z_type` value of 'Armadillo' sorts first and that `z_type` value of 'armadillo' sorts after 'Zebra'.

```
Select
  z_type As "Type"
, z_cost AS "Price"
, z_name As "Name"
From zoo_2016
ORDER BY z_type, z_cost;
```

Type	Price	Name
Armadillo	490	Abigail
Giraffe	3750	Dewey
Giraffe	5000	Sam
Giraffe	5000	Geoff
Giraffe	5000	
Giraffe	5000	
Giraffe	5000	Arnold
Giraffe	5000.25	Sally Robinson
Horse	490	
Lion	1850	
Lion	1850	
Lion	1850	
Lion	1850	Leon
Lion	5000	Lenora
Lion	5000	Leon
Zebra	2500.25	Huey
Zebra	2500.25	Dewey
Zebra	2500.25	Louie
armadillo	490	Anders
armadillo	490.01	Anne

Demo 09: This is a two level sort. The first sort key is the `z_type` and it is ascending. The second sort key `z_cost` uses a descending sort.

```
Select
  z_type As "Type"
, z_cost AS "Price"
, z_name As "Name"
From zoo_2016
ORDER BY z_type, z_cost desc;
```

Type	Price	Name
Armadillo	490	Abigail
Giraffe	5000.25	Sally Robinson
Giraffe	5000	Sam
Giraffe	5000	Geoff
Giraffe	5000	
Giraffe	5000	
Giraffe	5000	Arnold
Giraffe	3750	Dewey
Horse	490	
Lion	5000	Leon
Lion	5000	Lenora
Lion	1850	

Lion	1850	
Lion	1850	Leon
Lion	1850	
Zebra	2500.25	Huey
Zebra	2500.25	Louie
Zebra	2500.25	Dewey
armadillo	490.01	Anne
armadillo	490	Anders

Demo 10: The Oracle default is that nulls sort as a high-valued data item. We have some animals with no name value. They are sorting at the end of this display

```

Select
  z_type As "Type"
, z_name As "Name"
From zoo_2016
ORDER BY z_name;
```

Type	Name
-----	-----
Armadillo	Abigail
armadillo	Anders
armadillo	Anne
Giraffe	Arnold
Giraffe	Dewey
Zebra	Dewey
Giraffe	Geoff
Zebra	Huey
Lion	Lenora
Lion	Leon
Lion	Leon
Zebra	Louie
Giraffe	Sally Robinson
Giraffe	Sam
Lion	
Giraffe	
Lion	
Horse	
Lion	
Giraffe	

Demo 11: You can specify a NULLS FIRST or NULLS LAST option. This is a NULLS FIRST sort and the nulls appear at the start of the result set.

```

select
  z_type As "Type"
, z_name As "Name"
from zoo_2016
ORDER BY z_name NULLS FIRST;
```

Type	Name
-----	-----
Lion	
Lion	
Lion	
Giraffe	
Giraffe	
Horse	
Armadillo	Abigail
armadillo	Anders
armadillo	Anne
Giraffe	Arnold
Giraffe	Dewey

Zebra	Dewey
Giraffe	Geoff
Zebra	Huey
Lion	Lenora
Lion	Leon
Lion	Leon
Zebra	Louie
Giraffe	Sally Robinson
Giraffe	Sam

Demo 12: Using a NULLS FIRST sort with the names in descending order. The Nulls First/Last option places the nulls at the start or at the end of the result set.

```
select
  z_type as "Type"
, z_name as "Name"
from zoo_2016
ORDER BY z_name DESC NULLS FIRST
;
```

Type	Name
Giraffe	
Lion	
Giraffe	
Lion	
Lion	
Horse	
Giraffe	Sam
Giraffe	Sally Robinson
Zebra	Louie
Lion	Leon
Lion	Leon
Lion	Lenora
Zebra	Huey
Giraffe	Geoff
Zebra	Dewey
Giraffe	Dewey
Giraffe	Arnold
armadillo	Anne
armadillo	Anders
Armadillo	Abigail

Demo 13: You can sort on a date value.

```
Select
  z_id
, z_dob as "BirthDate"
, z_name as "Name"
From zoo_2016
ORDER BY z_dob DESC
;
```

Z_ID	BirthDate	Name
47	10-JAN-15	
44	02-JUN-14	Dewey
23	15-MAY-14	Sam
85	15-MAY-14	Sally Robinson
257	15-MAY-14	Arnold
57	25-MAR-14	Lenora
52	06-JUN-13	Dewey

```

43 02-JUN-13 Huey
258 15-MAY-13
373 25-FEB-13
374 25-FEB-13
372 25-FEB-13 Leon
375 25-FEB-13
371 15-JAN-13 Anne
370 15-JAN-13 Anders
25 15-JAN-13 Abigail
45 02-JAN-13 Louie
56 25-FEB-10 Leon
260 15-MAY-02 Geoff
259 15-MAY-02

```

Demo 14: Oracle allows you to sort by an alias. But if this is a quoted alias, then the sort key must also be a quoted alias.

```

Select
  z_id
, z_dob as "Date of Birth"
, z_name as "Name"
From zoo_2016
ORDER BY "Date of Birth"
;

```

Z_ID	BirthDate	Name
259	15-MAY-02	
260	15-MAY-02	Geoff
56	25-FEB-10	Leon
45	02-JAN-13	Louie
25	15-JAN-13	Abigail
370	15-JAN-13	Anders
371	15-JAN-13	Anne
375	25-FEB-13	
372	25-FEB-13	Leon
374	25-FEB-13	
373	25-FEB-13	
258	15-MAY-13	
43	02-JUN-13	Huey
52	06-JUN-13	Dewey
57	25-MAR-14	Lenora
85	15-MAY-14	Sally Robinson
257	15-MAY-14	Arnold
23	15-MAY-14	Sam
44	02-JUN-14	Dewey
47	10-JAN-15	

Demo 15: Oracle allows you to sort by the column number. This is generally considered poor style since it is easy to rearrange the column in the select and forget to adjust the Order By clause. You want to write SQL that is easier to write correctly and harder to write incorrectly.
This will sort by the z_type values then by the z_name values.

```

Select
  z_id
, z_type
, z_name
From zoo_2016
ORDER BY 2,3
;

```


Z_ID	Z_TYPE	Z_NAME
25	Armadillo	Abigail
257	Giraffe	Arnold
52	Giraffe	Dewey
260	Giraffe	Geoff
85	Giraffe	Sally Robinson
23	Giraffe	Sam
258	Giraffe	
259	Giraffe	
47	Horse	
57	Lion	Lenora
56	Lion	Leon
372	Lion	Leon
374	Lion	
375	Lion	
373	Lion	
44	Zebra	Dewey
43	Zebra	Huey
45	Zebra	Louie
370	armadillo	Anders
371	armadillo	Anne

Demo 16: You can sort on calculated columns, either by using the alias or repeating the calculation as the sort key. We discuss calculation later; this is included here for completeness. `extract (Month. . .)` returns the numerical value of the month.

```
select
  z_id
, extract ( month from z_dob ) as "Birth Month"
, z_dob
, z_name as "Name"
from zoo_2016
order by extract ( month from z_dob ) ;
```

Z_ID	Birth Month	Z_DOB	Name
45	1	02-JAN-13	Louie
25	1	15-JAN-13	Abigail
371	1	15-JAN-13	Anne
370	1	15-JAN-13	Anders
47	1	10-JAN-15	
56	2	25-FEB-10	Leon
374	2	25-FEB-13	
375	2	25-FEB-13	
373	2	25-FEB-13	
372	2	25-FEB-13	Leon
57	3	25-MAR-14	Lenora
258	5	15-MAY-13	
259	5	15-MAY-02	
260	5	15-MAY-02	Geoff
23	5	15-MAY-14	Sam
85	5	15-MAY-14	Sally Robinson
257	5	15-MAY-14	Arnold
52	6	06-JUN-13	Dewey
43	6	02-JUN-13	Huey
44	6	02-JUN-14	Dewey