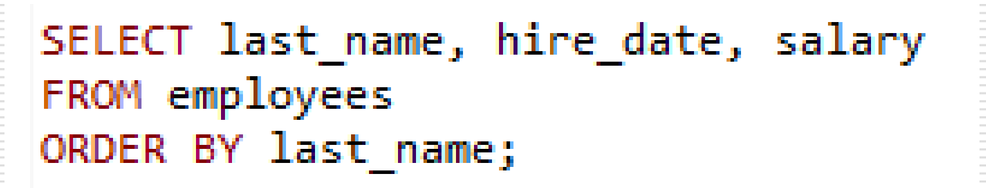
Sorting Rows with the ORDER BY Clause

ORDER BY Clause

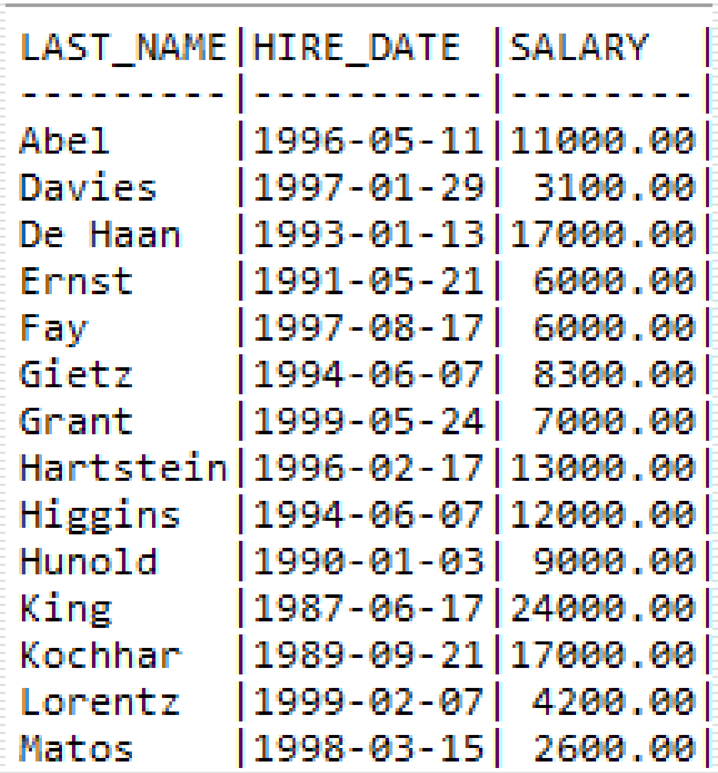
- Follows the FROM clause
- Last clause of the SQL statement
- Sort Sequence:
  - Ascending sequence is the default and is optional
  - DESC keyword is specified following the column name that is to be sorted in descending sequence

ORDER BY Clause – Character Values

- Character values are sorted in alphabetical order



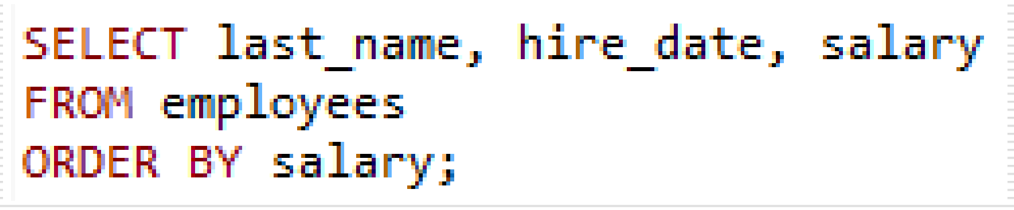
```sql
SELECT last_name, hire_date, salary
FROM employees
ORDER BY last_name;
```



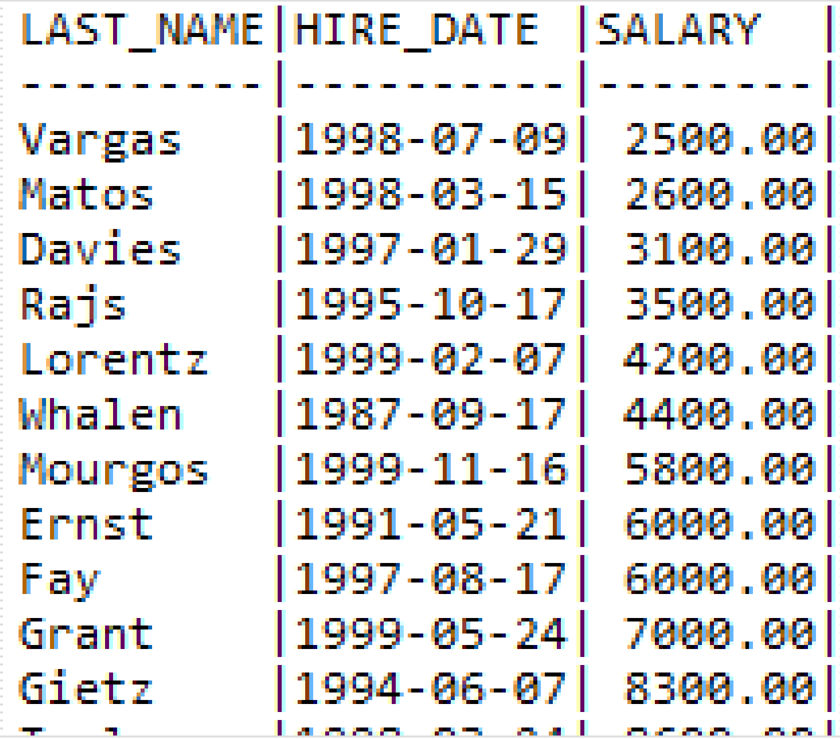
```
LAST_NAME|HIRE_DATE |SALARY   |
---------|----------|--------|
Abel     |1996-05-11|11000.00|
Davies   |1997-01-29| 3100.00|
De Haan  |1993-01-13|17000.00|
Ernst    |1991-05-21| 6000.00|
Fay      |1997-08-17| 6000.00|
Gietz    |1994-06-07| 8300.00|
Grant    |1999-05-24| 7000.00|
Hartstein|1996-02-17|13000.00|
Higgins  |1994-06-07|12000.00|
Hunold   |1990-01-03| 9000.00|
King     |1987-06-17|24000.00|
Kochhar  |1989-09-21|17000.00|
Lorentz  |1999-02-07| 4200.00|
Matos    |1998-03-15| 2600.00|
```

ORDER BY Clause – Numeric Values

- Numeric values are sorted lowest to highest



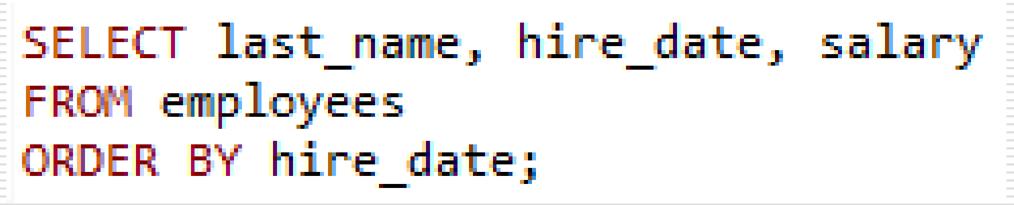
```
SELECT last_name, hire_date, salary
FROM employees
ORDER BY salary;
```



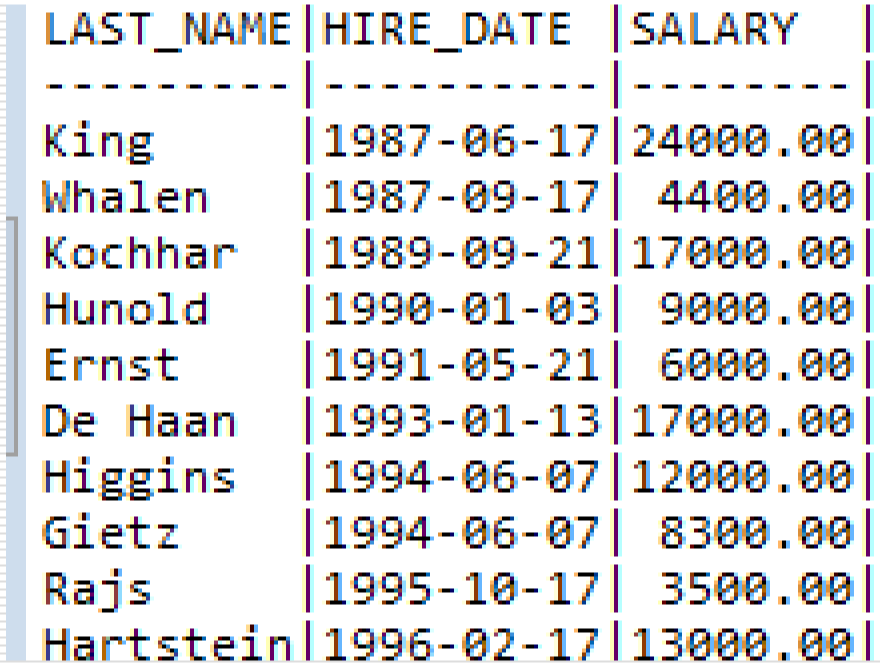
```
LAST_NAME |HIRE_DATE  |SALARY   |
----------|-----------|---------|
Vargas    |1998-07-09| 2500.00|
Matos     |1998-03-15| 2600.00|
Davies    |1997-01-29| 3100.00|
Rajs      |1995-10-17| 3500.00|
Lorentz   |1999-02-07| 4200.00|
Whalen    |1987-09-17| 4400.00|
Mourgos   |1999-11-16| 5800.00|
Ernst     |1991-05-21| 6000.00|
Fay       |1997-08-17| 6000.00|
Grant     |1999-05-24| 7000.00|
Gietz     |1994-06-07| 8300.00|
```

ORDER BY Clause – Date Values

- Date values are displayed with the earliest value first

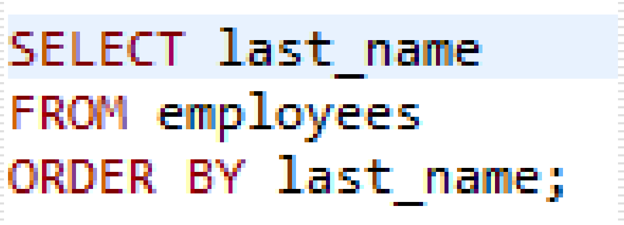


```sql
SELECT last_name, hire_date, salary
FROM employees
ORDER BY hire_date;
```

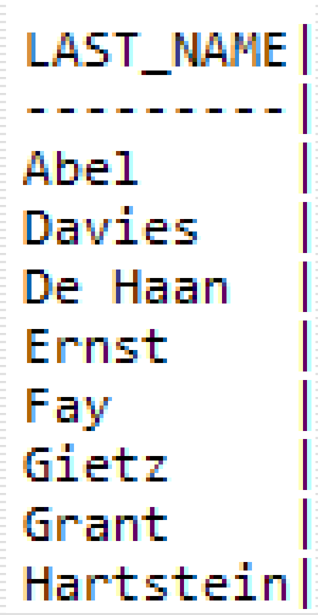


```
LAST_NAME|HIRE_DATE |SALARY  |
---------|----------|--------|
King     |1987-06-17|24000.00|
Whalen   |1987-09-17| 4400.00|
Kochhar  |1989-09-21|17000.00|
Hunold   |1990-01-03| 9000.00|
Ernst    |1991-05-21| 6000.00|
De Haan  |1993-01-13|17000.00|
Higgins  |1994-06-07|12000.00|
Gietz    |1994-06-07| 8300.00|
Rajs     |1995-10-17| 3500.00|
Hartstein|1996-02-17|13000.00|
```

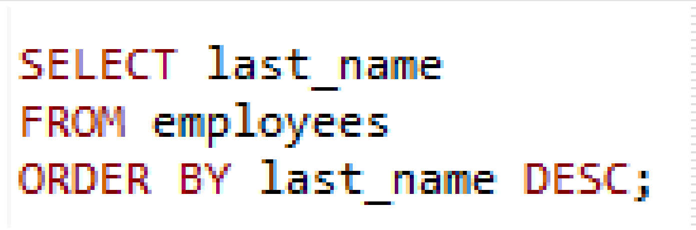
ORDER BY Clause – Descending Sequence



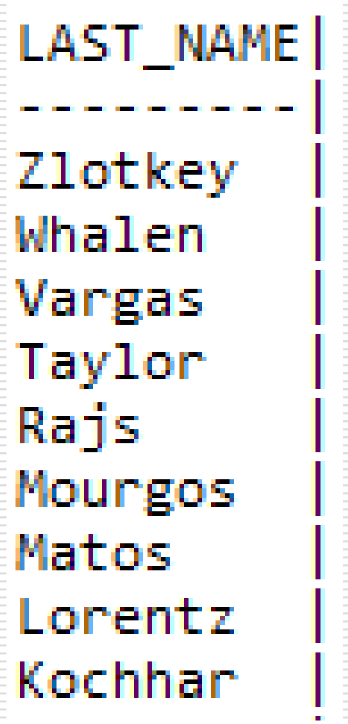
```sql
SELECT last_name
FROM employees
ORDER BY last_name;
```



```
LAST_NAME|
---------|
Abel     |
Davies   |
De Haan  |
Ernst    |
Fay      |
Gietz    |
Grant    |
Hartstein|
```



```sql
SELECT last_name
FROM employees
ORDER BY last_name DESC;
```



```
LAST_NAME|
---------|
Zlotkey  |
Whalen   |
Vargas   |
Taylor   |
Rajs     |
Mourgos  |
Matos    |
Lorentz  |
Kochhar  |
```

Sorting on Multiple Columns

Example 9-27

Return city, credit limit, and company name for companies with a credit limit greater than 140000. Sort the result set credit limit within city.

Data Set ds9_27

```
ID |CUSTOMER_NAME   |CITY     |CREDIT_LIMIT|
---|----------------|---------|------------|
101|Bargains Galore|Detroit  |      125000|
102|RedHot Discount|San Diego|      185500|
103|NewTown Deals  |Dallas   |      132500|
104|Ajax Sales     |Detroit  |      150000|
105|BigBox Direct  |Chicago  |      145000|
106|Mainstreet Inc |Dallas   |      200000|
107|Riverside Mfg  |Chicago  |      164000|
108|Cube Industries|Dallas   |      185000|
109|LowCost Shops  |San Diego|      155000|
110|Sterling Mfg   |Chicago  |      180000|
```

SQL Statement

```
SELECT city, credit_limit, customer_name
FROM   ds9_27
WHERE  credit_limit > 140000
ORDER BY city, credit_limit;
```

Result Set

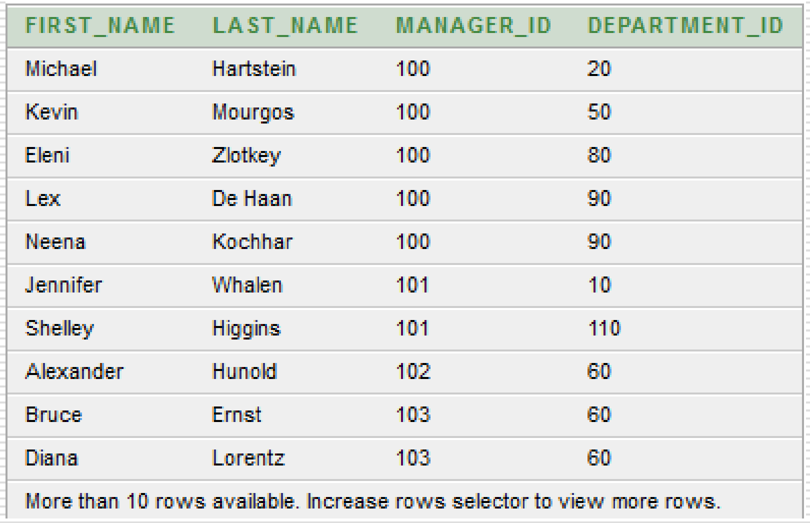
```
CITY     |CREDIT_LIMIT|CUSTOMER_NAME  |
---------|------------|---------------|
Chicago  |      145000|BigBox Direct  |
Chicago  |      164000|Riverside Mfg  |
Chicago  |      180000|Sterling Mfg   |
Dallas   |      185000|Cube Industries|
Dallas   |      200000|Mainstreet Inc |
Detroit  |      150000|Ajax Sales     |
San Diego|      155000|LowCost Shops  |
San Diego|      185500|RedHot Discount|
```

Sorting with Multiple Columns

```
SELECT first_name, last_name, manager_id, department_id
  FROM employees
  ORDER BY manager_id, department_id, last_name;
```

- Minor to Major sorts

1. Sort last_name (alphabetical order A to Z)
2. Sort department_id (lowest to highest)
3. Sort manager_id (lowest to highest)



FIRST_NAME	LAST_NAME	MANAGER_ID	DEPARTMENT_ID
Michael	Hartstein	100	20
Kevin	Mourgos	100	50
Eleni	Zlotkey	100	80
Lex	De Haan	100	90
Neena	Kochhar	100	90
Jennifer	Whalen	101	10
Shelley	Higgins	101	110
Alexander	Hunold	102	60
Bruce	Ernst	103	60
Diana	Lorentz	103	60

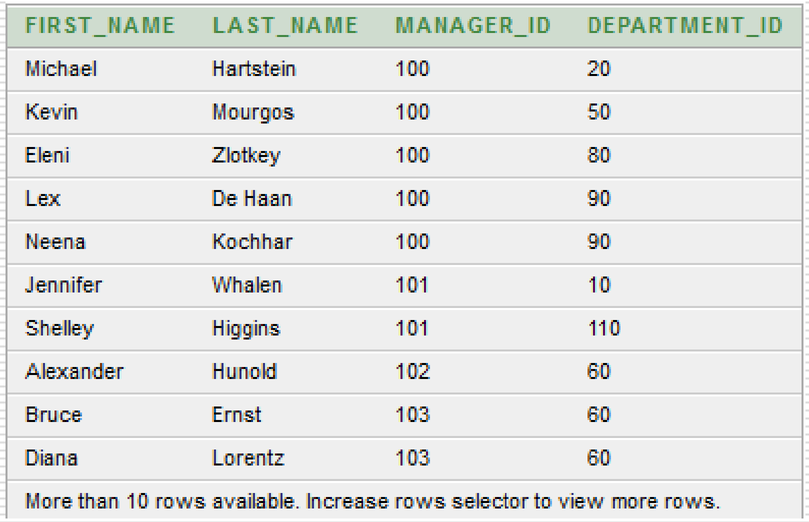
More than 10 rows available. Increase rows selector to view more rows.

Sorting with Multiple Columns Another Way to Say It

```
SELECT first_name, last_name, manager_id, department_id
  FROM employees
  ORDER BY manager_id, department_id, last_name;
```

- Minor to Major sorts

1. Sort last_name (alphabetical order A to Z)
2. Within department_id (lowest to highest)
3. Within manager_id (lowest to highest)



FIRST_NAME	LAST_NAME	MANAGER_ID	DEPARTMENT_ID
Michael	Hartstein	100	20
Kevin	Mourgos	100	50
Eleni	Zlotkey	100	80
Lex	De Haan	100	90
Neena	Kochhar	100	90
Jennifer	Whalen	101	10
Shelley	Higgins	101	110
Alexander	Hunold	102	60
Bruce	Ernst	103	60
Diana	Lorentz	103	60

More than 10 rows available. Increase rows selector to view more rows.

Sorting DESC on Multiple Columns

Example 9-28

Return city, credit limit, and company name for companies with a credit limit greater than 140000. Sort the result set credit limit (descending) within city (descending)

Data Set ds9_28

```
ID |CUSTOMER_NAME   |CITY     |CREDIT_LIMIT|
---|----------------|---------|------------|
101|Bargains Galore|Detroit  |      125000|
102|RedHot Discount|San Diego|      185500|
103|NewTown Deals  |Dallas   |      132500|
104|Ajax Sales     |Detroit  |      150000|
105|BigBox Direct  |Chicago  |      145000|
106|Mainstreet Inc |Dallas   |      200000|
107|Riverside Mfg  |Chicago  |      164000|
108|Cube Industries|Dallas   |      185000|
109|LowCost Shops  |San Diego|      155000|
110|Sterling Mfg   |Chicago  |      180000|
```

SQL Statement

```
SELECT city, credit_limit, customer_name
FROM   ds9_28
WHERE  credit_limit > 140000
ORDER BY city, credit_limit DESC;
```

Result Set

```
CITY     |CREDIT_LIMIT|CUSTOMER_NAME   |
---------|------------|----------------|
Chicago  |      180000|Sterling Mfg    |
Chicago  |      164000|Riverside Mfg   |
Chicago  |      145000|BigBox Direct   |
Dallas   |      200000|Mainstreet Inc  |
Dallas   |      185000|Cube Industries|
Detroit  |      150000|Ajax Sales      |
San Diego|      185500|RedHot Discount|
San Diego|      155000|LowCost Shops   |
```

Sorting DESC with Multiple Columns

- Reverse the sort order of a column by adding DESC after its name

```
SELECT department_id, last_name
FROM employees
WHERE department_id <= 50
ORDER BY department_id DESC, last_name;
```

DEPARTMENT_ID	LAST_NAME
50	Davies
50	Matos
50	Mourgos
50	Rajs
50	Vargas
20	Fay
20	Hartstein
10	Whalen

Using a Relative Column Number

Example 9-29 Using the ORDER BY clause with a relative column number

Data Set ds9_29

```
ID |CUSTOMER_NAME   |CITY     |CREDIT_LIMIT|
---|----------------|---------|------------|
101|Bargains Galore|Detroit  |      125000|
102|RedHot Discount|San Diego|      185500|
103|NewTown Deals   |Dallas   |      132500|
104|Ajax Sales      |Detroit  |      150000|
105|BigBox Direct   |Chicago  |      145000|
106|Mainstreet Inc  |Dallas   |      200000|
107|Riverside Mfg   |Chicago  |      164000|
108|Cube Industries|Dallas   |      185000|
109|LowCost Shops   |San Diego|      155000|
110|Sterling Mfg    |Chicago  |      180000|
```

SQL Statement

```
SELECT city, credit_limit, customer_name
FROM   ds9_29
WHERE  credit_limit > 140000
ORDER BY city, 2 DESC;
```

Result Set

```
CITY     |CREDIT_LIMIT|CUSTOMER_NAME   |
---------|------------|----------------|
Chicago  |      180000|Sterling Mfg    |
Chicago  |      164000|Riverside Mfg   |
Chicago  |      145000|BigBox Direct   |
Dallas   |      200000|Mainstreet Inc  |
Dallas   |      185000|Cube Industries|
Detroit  |      150000|Ajax Sales      |
San Diego|      185500|RedHot Discount|
San Diego|      155000|LowCost Shops   |
```

Using Column Alias on ORDER BY Clause

Example 9-31 Using an alias name on the ORDER BY clause for a calculated column

Data Set ds9_31

```
ID |CUSTOMER_NAME   |CITY     |CREDIT_LIMIT|BALANCE  |
---|---------------|---------|------------|---------|
101|Bargains Galore|Detroit  |      125000|110083.00|
102|RedHot Discount|San Diego|      185500|120387.00|
103|NewTown Deals  |Dallas   |      132500|109635.00|
104|Ajax Sales     |Detroit  |      150000|118630.00|
105|BigBox Direct  |Chicago  |      145000|123122.00|
106|Mainstreet Inc |Dallas   |      200000|116588.00|
107|Riverside Mfg  |Chicago  |      164000|110230.00|
108|Cube Industries|Dallas   |      185000|119525.00|
109|LowCost Shops  |San Diego|      155000|106687.00|
110|Sterling Mfg   |Chicago  |      180000|122429.00|
```

SQL Statement

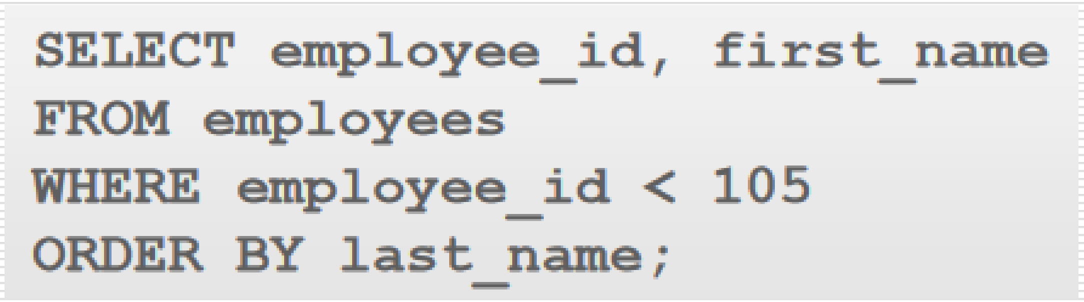
```sql
SELECT city AS "City",
       customer_name AS "Customer",
       credit_limit - balance AS "Available Credit"
FROM   ds9_31
WHERE  credit_limit - balance > 50000
ORDER BY city, "Available Credit" DESC;
```

Result Set

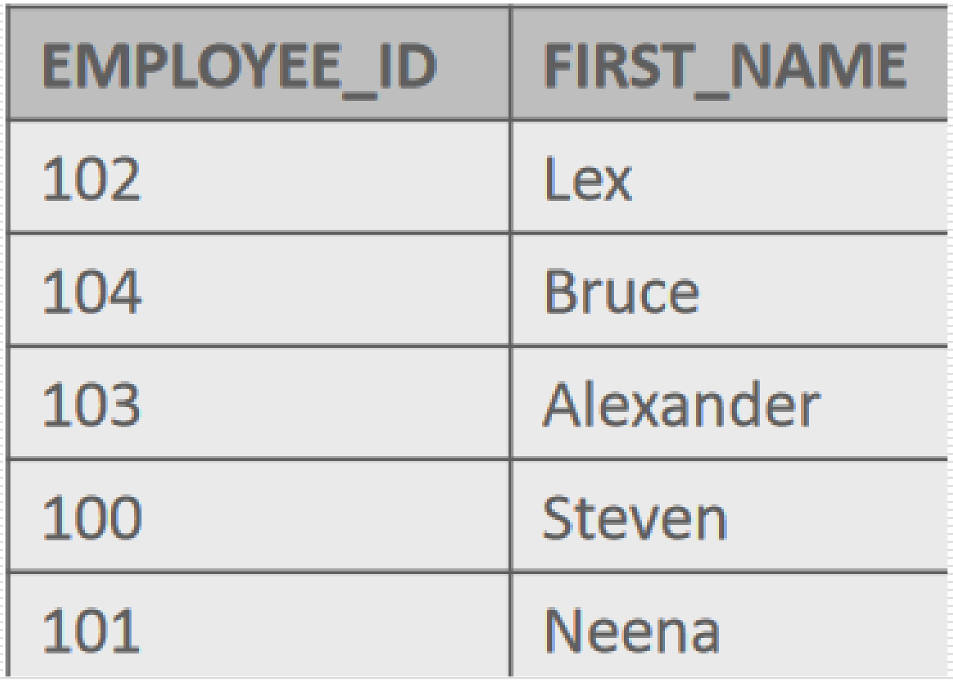
```
City     |Customer       |Available Credit|
---------|---------------|----------------|
Chicago  |Sterling Mfg   |        57571.00|
Chicago  |Riverside Mfg  |        53770.00|
Dallas   |Mainstreet Inc |        83412.00|
Dallas   |Cube Industries|        65475.00|
San Diego|RedHot Discount|        65113.00|
```

Sorting on Other Columns

- Order output by a column that is not listed in the SELECT clause



```
SELECT employee_id, first_name
FROM employees
WHERE employee_id < 105
ORDER BY last_name;
```

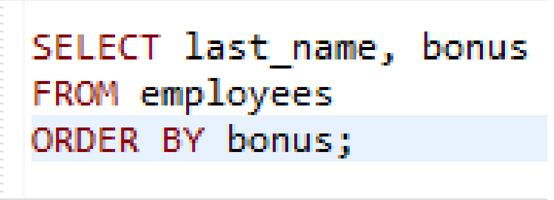


EMPLOYEE_ID	FIRST_NAME
102	Lex
104	Bruce
103	Alexander
100	Steven
101	Neena

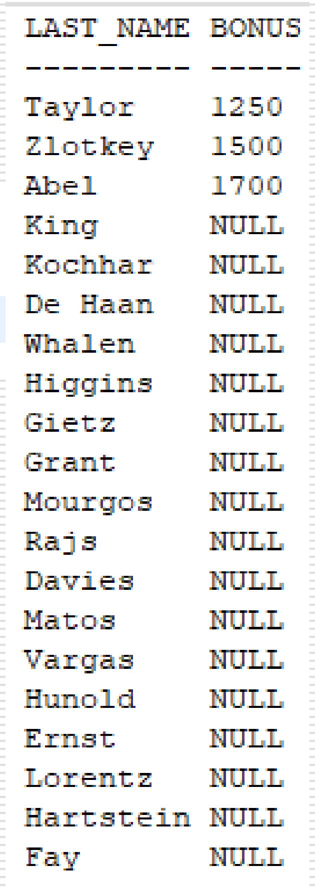
ORDER BY Clause – NULL Values

- NULL values are displayed last in ascending order and first in descending order

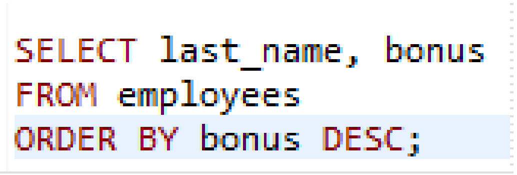
ORDER BY Clause – NULL Values



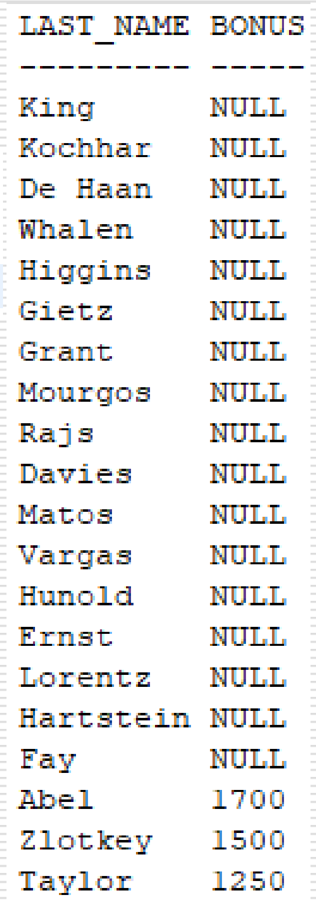
```
SELECT last_name, bonus
FROM employees
ORDER BY bonus;
```



```
LAST_NAME BONUS
--------- -----
Taylor    1250
Zlotkey   1500
Abel      1700
King      NULL
Kochhar   NULL
De Haan   NULL
Whalen    NULL
Higgins   NULL
Gietz     NULL
Grant     NULL
Mourgos   NULL
Rajs      NULL
Davies    NULL
Matos     NULL
Vargas    NULL
Hunold    NULL
Ernst     NULL
Lorentz   NULL
Hartstein NULL
Fay       NULL
```



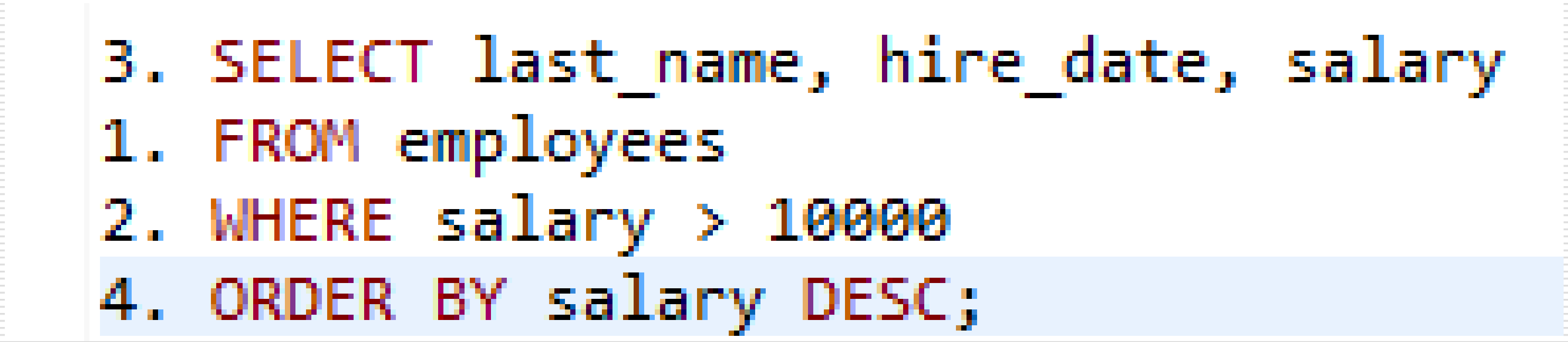
```
SELECT last_name, bonus
FROM employees
ORDER BY bonus DESC;
```



```
LAST_NAME BONUS
--------- -----
King      NULL
Kochhar   NULL
De Haan   NULL
Whalen    NULL
Higgins   NULL
Gietz     NULL
Grant     NULL
Mourgos   NULL
Rajs      NULL
Davies    NULL
Matos     NULL
Vargas    NULL
Hunold    NULL
Ernst     NULL
Lorentz   NULL
Hartstein NULL
Fay       NULL
Abel      1700
Zlotkey   1500
Taylor    1250
```

Order of Execution – Example 1

- Basic SELECT



```
3. SELECT last_name, hire_date, salary
1. FROM employees
2. WHERE salary > 10000
4. ORDER BY salary DESC;
```

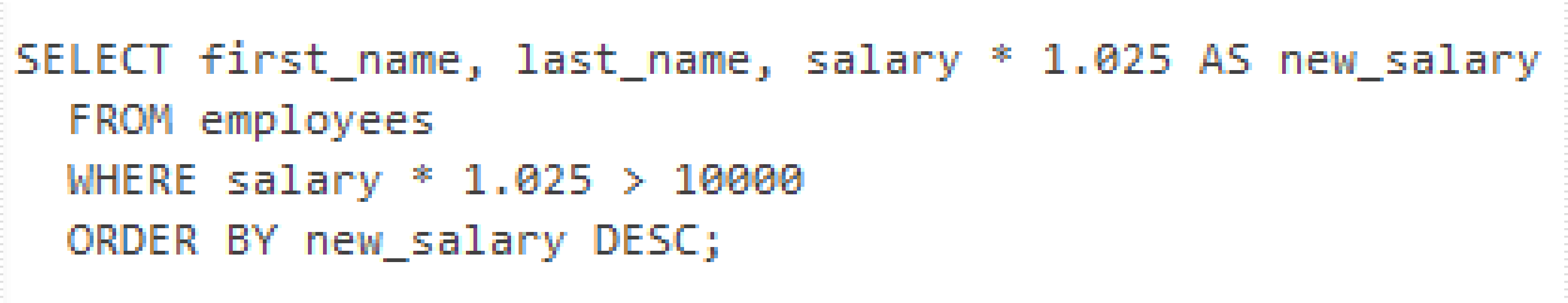
Order of Execution Example 2

- Give employees a 2.5% raise
- Set employees with a new salary > 10000
  - Use alias on WHERE and ORDER BY Clauses
  - What happens?

```
SELECT first_name, last_name, salary * 1.025 AS new_salary
  FROM employees
  WHERE new_salary > 10000
  ORDER BY new_salary DESC;
```

Order of Execution – Example 3

- Alias can be used on ORDER BY clause, but not WHERE clause
  - Why?
- Use the expression on the WHERE clause



```
SELECT first_name, last_name, salary * 1.025 AS new_salary
  FROM employees
  WHERE salary * 1.025 > 10000
  ORDER BY new_salary DESC;
```

SQL Statement Order of Execution

Order	Clause	Function
1	FROM	Connects to the database table
2	WHERE	Filters or restricts rows from the result set based on the criteria specified
3	SELECT	Filters or restricts columns from the result set based on the column-list
4	ORDER BY	Sorts the final result set

Q & A
Questions & Answers