

Retrieving Data Using the SQL `SELECT` Statement

Lesson Objectives

- After completing this lesson, you should be able to do the following:
 - List the capabilities of SQL `SELECT` statement
 - Execute a basic `SELECT` statement

ORACLE

Copyright © 2007, Oracle. All rights reserved.

1 - 2

Copyright © 2007, Oracle. All rights reserved.

ORACLE

Lesson Agenda

- Basic `SELECT` statement
- Arithmetic expressions and `NULL` values in the `SELECT` statement
- Column aliases
- Use of concatenation operator, literal character strings, alternative quote operator, and the `DISTINCT` keyword
- `DESCRIBE` command

ORACLE

1 - 3

Copyright © 2007, Oracle. All rights reserved.

Lesson Agenda

- **Basic `SELECT` statement**
- Arithmetic expressions and `NULL` values in the `SELECT` statement
- Column aliases
- Use of concatenation operator, literal character strings, alternative quote operator, and the `DISTINCT` keyword
- `DESCRIBE` command

ORACLE

1 - 4

Copyright © 2007, Oracle. All rights reserved.

Writing SQL Statements

- SQL statements are **not** case-sensitive.
- SQL statements can be entered on one or more lines.
- Keywords cannot be abbreviated or split across lines.
- In SQL Developer, SQL statements can optionally be terminated by a **semicolon (;)**. Semicolons are required when you execute multiple SQL statements.

ORACLE

1 - 5

Copyright © 2007, Oracle. All rights reserved.

Basic SELECT statement

```
SELECT * | {[DISTINCT] column|expression [alias],...}  
FROM table_name;
```

SELECT	a list of one or more columns
*	selects all columns
DISTINCT	suppresses duplicates
column expression	selects the named column or the expression
alias	gives the selected columns different headings
FROM table_name	specifies the table containing the columns

ORACLE

1 - 6

Copyright © 2007, Oracle. All rights reserved.

Selecting All Columns

- You can display all column of data in a table by following the **SELECT** keyword with **an asterisk (*)**.
- For example, the following SQL statement (like the example in the slide) displays all columns and all rows of the **DEPARTMENTS** table.

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	50	Shipping	124	1500
4	60	IT	103	1400
5	80	Sales	149	2500
6	90	Executive	100	1700

```
SELECT *  
FROM departments;
```

ORACLE

1 - 7

Copyright © 2007, Oracle. All rights reserved.

Selecting All Columns

Practice: The HR department wants a query to display all jobs.

	JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
1	AD_PRES	President	20000	40000
2	AD_VP	Administration Vice President	15000	30000
3	AD_ASST	Administration Assistant	3000	6000
4	FI_MGR	Finance Manager	8200	16000
5	FI_ACCOUNT	Accountant	4200	9000
6	AC_MGR	Accounting Manager	8200	16000
7	AC_ACCOUNT	Public Accountant	4200	9000
8	SA_MAN	Sales Manager	10000	20000
9	SA_REP	Sales Representative	6000	12000
10	PU_MAN	Purchasing Manager	8000	15000

ORACLE

1 - 8

Copyright © 2007, Oracle. All rights reserved.

Selecting Specific Columns

- You can use the `SELECT` statement to display specific columns of the table by specifying the column names, separated by commas.
- The example in the slide displays all the department numbers and location numbers from the `DEPARTMENTS` table.

	DEPARTMENT_ID	LOCATION_ID
1	10	1700
2	20	1800
3	50	1500
4	60	1400
5	80	2500
6	90	1700

```
SELECT department_id, location_id
FROM departments;
```

ORACLE

1 - 9

Copyright © 2007, Oracle. All rights reserved.

Selecting Specific Columns

Practice: Create report to display the employee id, first name, and last name for each employee.

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME
1	100	Steven	King
2	101	Neena	Kochhar
3	102	Lex	De Haan
4	103	Alexander	Hunold
5	104	Bruce	Ernst
6	105	David	Austin
7	106	Valli	Pataballa
8	107	Diana	Lorentz
9	108	Nancy	Greenberg
10	109	Daniel	Faviet

1 - 10

Copyright © 2007, Oracle. All rights reserved.

Column Heading Defaults

- SQL Developer:
 - Default heading alignment: Left-aligned
 - Default heading display: Uppercase

```
SELECT employee_id, first_name, last_name
FROM employees;
```

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME
1	100	Steven	King
2	101	Neena	Kochhar
3	102	Lex	De Haan
4	103	Alexander	Hunold
5	104	Bruce	Ernst
6	105	David	Austin
7	106	Valli	Pataballa
8	107	Diana	Lorentz



ORACLE

1 - 11

Copyright © 2007, Oracle. All rights reserved.

Lesson Agenda

- Basic `SELECT` statement
- Arithmetic expressions and NULL values in the SELECT statement**
- Column aliases
- Use to concatenation operator, literal character strings, alternative quote operator, and the `DISTINCT` keyword
- `DESCRIBE` command

ORACLE

1 - 12

Copyright © 2007, Oracle. All rights reserved.

Arithmetic Expressions

- Create expressions with number and date data by using arithmetic operators.

Operator	Description
+	Add
-	Subtract
*	Multiply
/	Divide

- Rules of Precedence:
 - Multiplication and division occur before addition and subtraction.
 - Operators of the same priority are evaluated from left to right.
 - Parentheses are used to override the default precedence or to clarify the statement.

ORACLE

1 - 13

Copyright © 2007, Oracle. All rights reserved.

Using Arithmetic Expressions

The example in the slide uses the addition operator to calculate a salary increase of \$300 for all employees. The slide also displays a SALARY + 300 column in the output.

	FIRST_NAME	SALARY	SALARY+300
1	Steven	24000	24300
2	Neena	17000	17300
3	Lex	17000	17300
4	Alexander	9000	9300
5	Bruce	6000	6300
6	David	4800	5100
7	Valli	4800	5100

```
SELECT first_name, salary, salary*300
FROM employees;
```

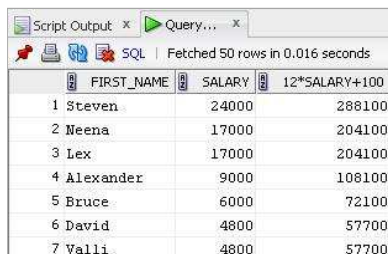
ORACLE

1 - 14

Copyright © 2007, Oracle. All rights reserved.

Operator Precedence

The first example in the slide display first_name, salary, and annual compensation of employees. It calculates the annual compensation by multiplying the monthly salary with 12, plus a one-time bonus of \$100.



	FIRST_NAME	SALARY	12*SALARY+100
1	Steven	24000	288100
2	Neena	17000	204100
3	Lex	17000	204100
4	Alexander	9000	108100
5	Bruce	6000	72100
6	David	4800	57700
7	Valli	4800	57700

Note that multiplication is performed before addition

1

```
SELECT first_name, salary, 12*salary+100
FROM employees;
```

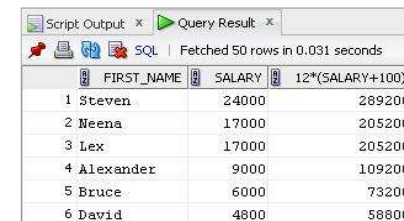
ORACLE

1 - 15

Copyright © 2007, Oracle. All rights reserved.

Operator Precedence Using Parentheses

The second example in the slide display first_name, salary, and annual compensation of employees. It calculates the annual compensation as follow: adding a monthly bonus \$100 to the monthly salary, and then multiplying that subtotal the monthly salary with 12.



	FIRST_NAME	SALARY	12*(SALARY+100)
1	Steven	24000	289200
2	Neena	17000	205200
3	Lex	17000	205200
4	Alexander	9000	109200
5	Bruce	6000	73200
6	David	4800	58800

Because of the parentheses, addition takes priority over multiplication

2

```
SELECT first_name, salary, 12*(salary+100)
FROM employees;
```

ORACLE

1 - 16

Copyright © 2007, Oracle. All rights reserved.

Using Arithmetic Operators

Practice: Display job id, job title, and minimum salary. Calculates the minimum salary by increased with 5% for each job.

	JOB_ID	JOB_TITLE	MIN_SALARY	Min Salary
1	AD_PRES	President	20000	21000
2	AD_VP	Administration Vice President	15000	15750
3	AD_ASST	Administration Assistant	3000	3150
4	FI_MGR	Finance Manager	8200	8610
5	FI_ACCOUNT	Accountant	4200	4410
6	AC_MGR	Accounting Manager	8200	8610
7	AC_ACCOUNT	Public Accountant	4200	4410

ORACLE

Defining a Null Value

- NULL is a value that is unavailable, unassigned, unknown, or inapplicable.
- NULL is not the same as zero or a blank space.

ORACLE

Defining a Null Value

In the COMMISSION_PCT column in the EMPLOYEES table, notice that only a sales manager or sales representative can earn a commission. Other employees are not entitled to earn commission null represents that fact.

```
SELECT last_name, job_id, salary, commission_pct
FROM employees;
```

	LAST_NAME	JOB_ID	SALARY	COMMISSION_PCT
1	King	AD_PRES	24000	(null)
2	Kochhar	AD_VP	17000	(null)

...

12	Zlotkey	SA_MAN	10500	0.2
13	Abel	SA_REP	11000	0.3
14	Taylor	SA_REP	8600	0.2

ORACLE

Null Values in Arithmetic Expressions

- Arithmetic expressions containing a null value evaluate to null.
- In the example in the slide, employee King does not get any commission. Because the COMMISSION_PCT column in the arithmetic expression is null, the result is null.

```
SELECT last_name, job_id, salary,
       12*salary*commission_pct
FROM employees;
```

	LAST_NAME	12*SALARY*COMMISSION_PCT
1	King	(null)
2	Kochhar	(null)



...

12	Zlotkey	25200
13	Abel	39600
14	Taylor	20640

ORACLE

Lesson Agenda

- Basic `SELECT` statement
- Arithmetic expressions and `NULL` values in the `SELECT` statement
- **Column aliases**
- Use to concatenation operator, literal character strings, alternative quote operator, and the `DISTINCT` keyword
- `DESCRIBE` command

ORACLE

1 - 21

Copyright © 2007, Oracle. All rights reserved.

Defining a Column Alias

- Renames a column heading
- Is useful with calculations
- Immediately follows the column name (There can also be the optional **AS** keyword between the column name and alias.)
- Requires double quotation marks if it contains spaces or special characters, or if it is case-sensitive
- By default, alias headings appear in uppercase. If the alias contains spaces or special characters (such as # or \$), or if it is case-sensitive, enclose the alias in double quotation marks (" ").

ORACLE

1 - 22

Copyright © 2007, Oracle. All rights reserved.

Using Column Aliases

The first example displays the names and the commission percentages of all the employees.

```
SELECT last_name AS name, commission_pct comm
FROM employees;
```

	1	NAME	2	COMM
1	King			(null)
2	Kochhar			(null)
3	De Haan			(null)

...

Note:

- the optional **AS** keyword has been used before the column alias name.
- the SQL statement has the column aliases, **name** and **comm**, in lowercase, whereas the result of the query displays the column heading in uppercase.

ORACLE

1 - 23

Copyright © 2007, Oracle. All rights reserved.

Using Column Aliases

The second example displays the last names and annual salaries of all the employees.

```
SELECT last_name "Name",
       salary*12 "Annual Salary"
FROM employees;
```

	1	Name	2	Annual Salary
1	King			288000
2	Kochhar			204000
3	De Haan			204000

...

Note:

- Column alias **Annual Salary** contains a space and column alias **Name** displays the column heading in uppercase, it has been enclosed in double quotation marks.
- The column heading in the output is exactly the same as the column alias.

ORACLE

1 - 24

Copyright © 2007, Oracle. All rights reserved.

Using Column Aliases

Practice: Display employee id, employee name, and calculates a salary per year for all employees. Name the column headings **CODE**, **Emp Name**, and **Annual Salary**

	CODE	Emp Name	Annual Salary
1	100	Steven	288000
2	101	Neena	204000
3	102	Lex	204000
4	103	Alexander	108000
5	104	Bruce	72000

ORACLE

1 - 25

Copyright © 2007, Oracle. All rights reserved.

Lesson Agenda

- Basic `SELECT` statement
- Arithmetic expressions and `NULL` values in the `SELECT` statement
- Column aliases
- **Use to Concatenation operator, Literal character strings, Alternative quote operator, and the `DISTINCT` keyword**
- `DESCRIBE` command

ORACLE

1 - 26

Copyright © 2007, Oracle. All rights reserved.

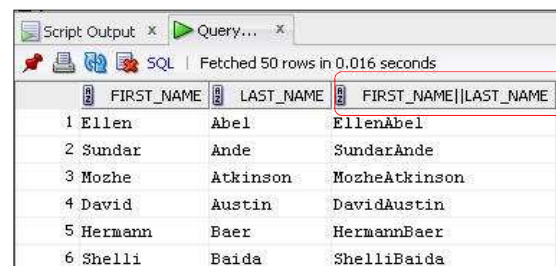
Concatenation Operator

- A concatenation operator:
 - Links columns or character strings to other columns
 - Is represented by **two vertical bars (||)**
 - Creates a resultant column that is a character expression

Concatenation Operator

In the example, first_name and last_name are concatenated

```
SELECT first_name, last_name,  
       first_name||last_name  
FROM   employees;
```



	FIRST_NAME	LAST_NAME	FIRST_NAME LAST_NAME
1	Ellen	Abel	EllenAbel
2	Sundar	Ande	SundarAnde
3	Mozhe	Atkinson	MozheAtkinson
4	David	Austin	DavidAustin
5	Hermann	Baer	HermannBaer
6	Shelli	Baida	ShelliBaida

ORACLE

1 - 27

Copyright © 2007, Oracle. All rights reserved.

ORACLE

1 - 28

Copyright © 2007, Oracle. All rights reserved.

Concatenation Operator

Practice: Display location id, street address, postal code, and the city concatenated with the state province of all locations.

	LOCATION_ID	STREET_ADDRESS	POSTAL_CODE	CITY
1	10001297	Via Cola di Rie	00989	Roma
2	110093091	Calle della Testa	10934	Venice
3	12002017	Shinjuku-ku	1689	Tokyo Tokyo Prefecture
4	13009450	Kamiya-cho	6823	Hiroshima
5	14002014	Jabberwocky Rd	26192	Southlake Texas
6	15002011	Interiors Blvd	99236	South San Francisco California
7	16002007	Zagora St	50000	South Brunswick New Jersey

--

ORACLE

1 - 29

Copyright © 2007, Oracle. All rights reserved.

Literal Character Strings

- A literal is a character, a number, or a date that is included in the `SELECT` statement.
- Date and character literal values must be enclosed within **single quotation marks(' ')**.
- Each character string is output once for each row returned.

ORACLE

1 - 30

Copyright © 2007, Oracle. All rights reserved.

Using Literal Character Strings

In the example, displays the last names and job codes of all employees. The column has the heading Employee Details.

```
SELECT last_name || ' is a ' || job_id
       AS "Employee Details"
FROM   employees;
```

	Employee Details
1	Abel is a SA_REP
2	Davies is a ST_CLERK
3	De Haan is a AD_VP
4	Ernst is a IT_PROG
5	Fay is a MK_REP

...

ORACLE

1 - 31

Copyright © 2007, Oracle. All rights reserved.

Using Literal Character Strings

Practice: Display the last name and salary for each employees are concatenated with a literal, to give the returned rows more meaning:

	MONTHLY
1	King: 1 Month salary = 24000
2	Kochhar: 1 Month salary = 17000
3	De Haan: 1 Month salary = 17000
4	Hunold: 1 Month salary = 9000
5	Ernst: 1 Month salary = 6000
6	Austin: 1 Month salary = 4800
7	Pataballa: 1 Month salary = 4800
8	Lorentz: 1 Month salary = 4200

--

ORACLE

1 - 32

Copyright © 2007, Oracle. All rights reserved.

Practice before study

1	Administration is manage bv:200
2	Marketing is manage bv:201
3	Purchasing is manage bv:114
4	Human Resources is manage bv:203
5	Shipping is manage bv:121
6	IT is manage bv:103
7	Public Relations is manage bv:204
8	Sales is manage bv:145
9	Executive is manage bv:100
10	Finance is manage bv:108
11	Accounting is manage bv:205
12	Treasury is manage bv:
13	Corporate Tax is manage bv:
14	Control And Credit is manage bv:
15	Shareholder Services is manage bv:

ORACLE

Alternative Quote (q) Operator

- Specify your own quotation mark delimiter.
- Select any delimiter.
- Increase readability and usability.

1	Administration Department's Manager Id:200
2	Marketing Department's Manager Id:201
3	Shipping Department's Manager Id:124
4	IT Department's Manager Id:103
5	Sales Department's Manager Id:149
6	Executive Department's Manager Id:100
7	Accounting Department's Manager Id:205
8	Contracting Department's Manager Id:

ORACLE

Alternative Quote (q) Operator

Step 1 Concatenation Operator && Literal Character Strings

1	Department and Manager
Administration Department	Manager Id: 200
Marketing Department	Manager Id: 201
Purchasing Department	Manager Id: 114
Human Resources Department	Manager Id: 203
Shipping Department	Manager Id: 121
IT Department	Manager Id: 103
Public Relations Department	Manager Id: 204

Column Header

Literal String

```
SELECT department_name ||
        'Department Manager Id:' || manager_id
AS "Department and Manager"
FROM departments;
```

ORACLE

Alternative Quote (q) Operator

Step 2 Alternative Quote Operator, To display the data with the following results.

1	Administration Department's Manager Id:200
2	Marketing Department's Manager Id:201
3	Shipping Department's Manager Id:124
4	IT Department's Manager Id:103
5	Sales Department's Manager Id:149

```
SELECT department_name ||
        q'[Department's Manager Id:]' ||
        manager_id AS "Department and Manager"
FROM departments;
```

ORACLE

Alternative Quote (q) Operator

Practice: Create a report of all employees and their department ids, display the first name concatenated with department id. Separate each column output by literal "Employee's Department Id: ". Name the column title Employee and Department

Employee Name Department
Steven Employee's Department Id: 90
Neena Employee's Department Id: 90
Lex Employee's Department Id: 90
Alexander Employee's Department Id: 60

Duplicate Rows

The default display of queries is all rows, including duplicate rows.

```
SELECT department_id FROM employees;
```

1

DEPARTMENT_ID
1
2
3
4
5

```
SELECT DISTINCT department_id FROM employees;
```

2

DEPARTMENT_ID
1
2
3
4

Duplicate Rows

Practice: Display all unique department numbers and job ids from the EMPLOYEES table.

DEPARTMENT_ID	JOB_ID
1	110 AC_ACCOUNT
2	90 AD_VP
3	50 ST_CLERK
4	80 SA_REP
5	50 ST_MAN
6	80 SA_MAN
7	110 AC_MGR
8	90 AD_PRES
9	60 IT_PROG
10	100 FI_MGR
11	30 PU_CLERK

Lesson Agenda

- Basic `SELECT` statement
- Arithmetic expressions and `NULL` values in the `SELECT` statement
- Column aliases
- Use to *Concatenation operator*, *Literal character strings*, *Alternative quote operator*, and the *`DISTINCT` keyword*
- **`DESCRIBE` command**

Displaying the Table Structure

- Use the DESCRIBE command to display the structure of a table
- Or select the table in the Connection tree and use the Column tab to view the table structure.

```
DESC[RIBE] table_name
```

Using the DESCRIBE command

```
DESCRIBE employees
```

Name	Null?	Type
EMPLOYEE_ID	NOT NULL	NUMBER(6)
FIRST_NAME		VARCHAR2(20)
LAST_NAME	NOT NULL	VARCHAR2(25)
EMAIL	NOT NULL	VARCHAR2(25)
PHONE_NUMBER		VARCHAR2(20)
HIRE_DATE	NOT NULL	DATE
JOB_ID	NOT NULL	VARCHAR2(10)
SALARY		NUMBER(8,2)
COMMISSION_PCT		NUMBER(2,2)
MANAGER_ID		NUMBER(6)
DEPARTMENT_ID		NUMBER(4)

Data Type	Description
NUMBER(p,s)	Number value having a maximum number of digits p, with s digits to the right of the decimal point
VARCHAR2(s)	Variable-length character value of maximum size s
DATE	Date and time value between January 1,4712 B.C. and December 31,A.D. 9999.

Quiz

Identify the SELECT statements that execute successfully.

1 SELECT first_name,last_name,job_id,
salary*12 AS Yearly Sal
FROM employees;

2 SELECT first_name,last_name,job_id,
salary*12 yearly
FROM employees;

3 SELECT first_name,last_name,job_id,
salary AS yearlySal
FROM employees;

4 SELECT first_name+last_name AS name,
job_id,salary*12 yearly Sal
FROM employees;

Summary

- In this lesson, you should have learned how to:
 - Write a SELECT statement that:
 - Returns all rows and columns from a table
 - Returns specified columns from a table
 - Uses column aliases to display more descriptive column headings

```
SELECT *|{[DISTINCT] column|expression [alias],...}  
FROM table_name;
```