

SQL Essentials

Query Formulation Using SELECT Statements



Objectives

- List the capabilities of the SQL SELECT statement.
- Write select statements involving single tables.
- Learn how to execute simple SQL statements.
- Develop familiarity with the tools you will use to develop SQL statements.



SELECT Statement

SELECT statements includes:

- A SELECT clause specifying the columns to be displayed.
- A FROM clause which identifies the table containing the columns that are listed in the SELECT clause.



Syntax For A SELECT Statement

SELECT { * [DISTINCT] column|expression [alias],}
FROM table;

- **SELECT** – list of one or more columns.
- ***** – selects all columns.
- **DISTINCT** – removes duplicate values.
- **column/expression** – selects the named column or expression.
- **Alias** – gives different headings to the selected columns.
- **FROM** table – specifies the table containing the columns.



GLOSSARY

- **KEYWORD** – is an individual SQL element. Example the SELECT and FROM keywords.
- **STATEMENT** – a combination of two or more clauses for example SELECT * FROM DEPT
- **CLAUSE** – a part of an SQL statement for example select department_id, department_name.



Guidelines For Writing SQL Statements

Use the following Guidelines to construct valid statements that are easy to read and edit:

- SQL statements are not case sensitive.
- SQL statements can be entered on one or more lines.
- Keywords cannot be abbreviated or split across lines.
- Clauses are usually placed on separate lines.
- Indents are used to enhance readability.
- Semicolons are required when you execute multiple SQL statements.
- SQL statements are executed with a semicolon (;) or forward slash (/).

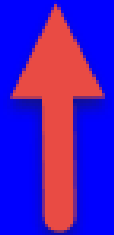
How To Execute An SQL Statement

SQL*Plus

- To execute SQL statements in SQL*Plus:
- Type a semicolon (;) at the end of the statement and press [Enter] on your keyboard to run the statement.

```
SQL> select employee_id, last_name, department_id from employees;
```

EMPLOYEE_ID	LAST_NAME	DEPARTMENT_ID
100	King	90
101	Kochhar	90
102	De Haan	90
103	Hunold	60
104	Ernst	60
105	Austin	60





How To Execute An SQL Statement

SQL*Plus

- Type the forward slash (/) at the end of the statement and press [Enter] on your keyboard to run the statement.

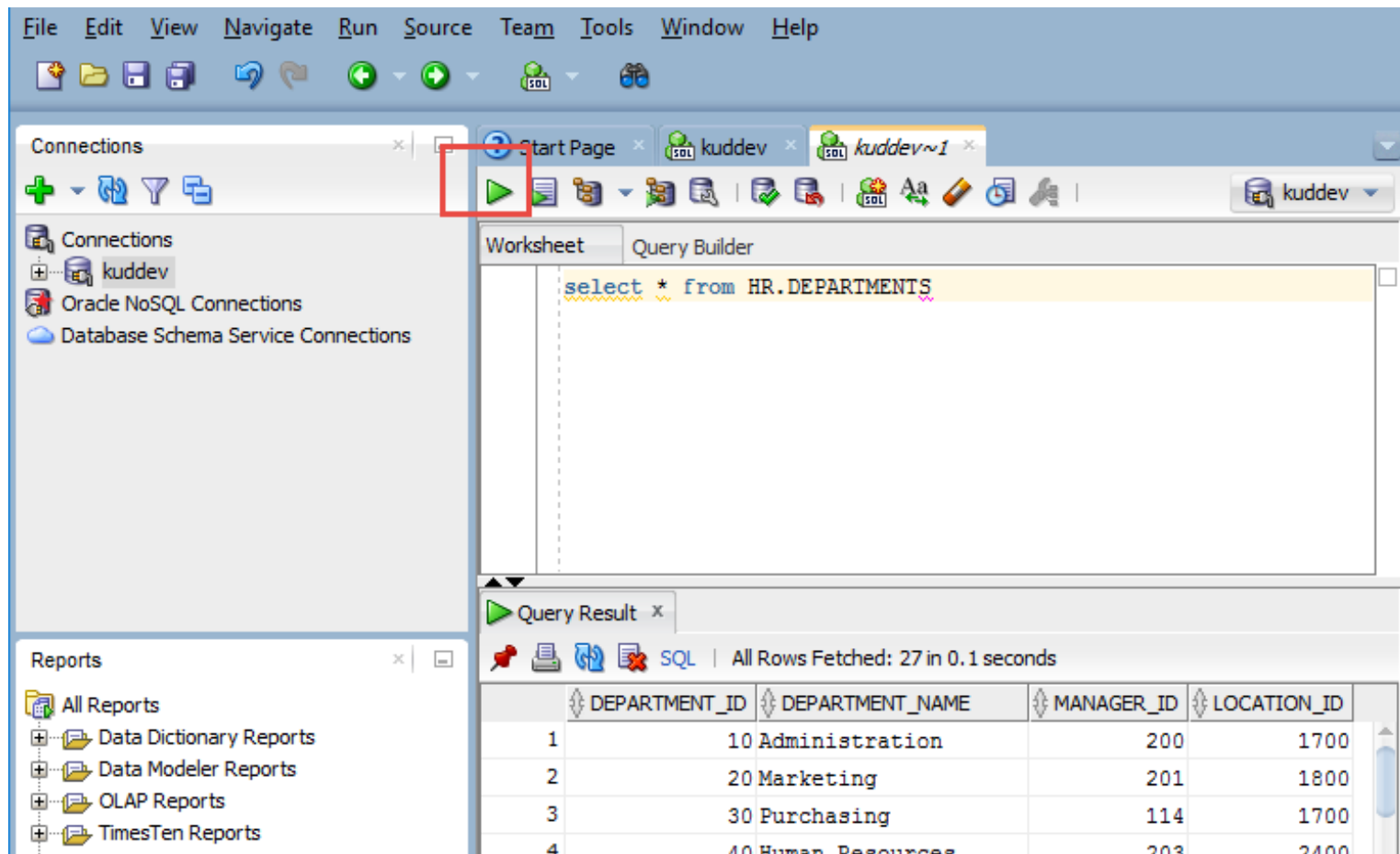
```
SQL> select department_id, department_name, manager_id from departments  
2 /
```



DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID
10	Administration	200
20	Marketing	201
30	Purchasing	114
40	Human Resources	203
50	Shipping	121
60	IT	103
70	Public Relations	204
80	Sales	145
90	Executive	100

How To Execute An SQL Statement

SQL Developer - To execute SQL statements in SQL Developer, click the **EXECUTE** statement icon or press [F9] to run the statement in the SQL worksheet.





Retrieve Data From All Columns of A Table

You can view all the columns of data in a table by using the asterix (*) in the SELECT clause.

SELECT * FROM table;

Example : To see all data in the departments tables type the following select statement.

SELECT * FROM departments;

Retrieve Data From All Columns of A Table

```
SELECT * FROM departments;
```

```
SQL> select * from departments;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
70	Public Relations	204	2700
80	Sales	145	2500
90	Executive	100	1700
100	Finance	108	1700
110	Accounting	205	1700

Retrieve Data From All Columns of A Table

You can also display all columns in a table by listing them after the SELECT keyword.

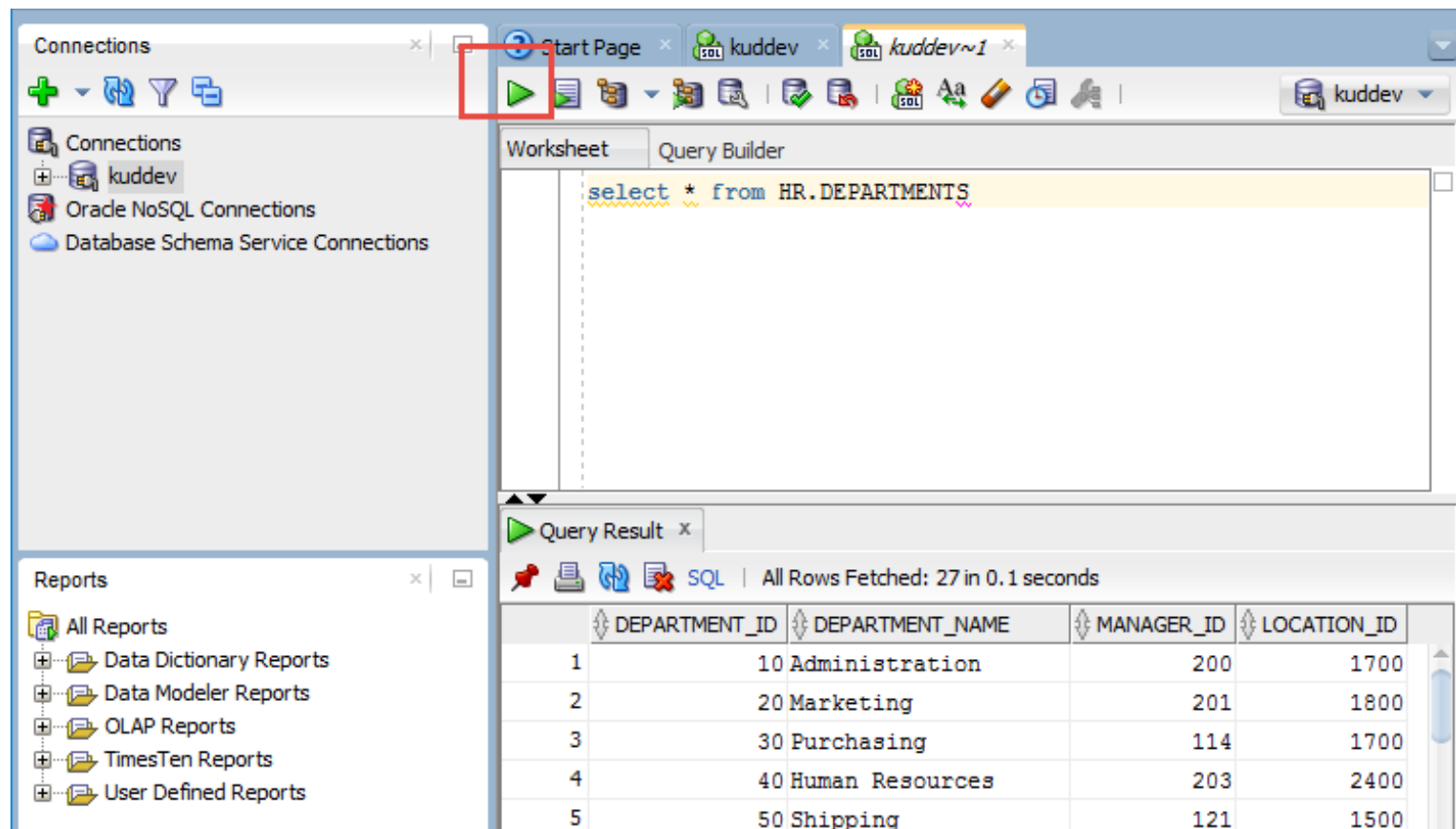
**SELECT department_id, department_name, manager_id, location_id
FROM departments;**

```
SQL> select department_id, department_name, manager_id, location_id  
2  from departments;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
30	Purchasing	114	1700
40	Human Resources	203	2400
50	Shipping	121	1500
60	IT	103	1400
70	Public Relations	204	2700
80	Sales	145	2500
90	Executive	100	1700
100	Finance	108	1700
110	Accounting	205	1700

Retrieve Data From All Columns of A Table

In SQL Developer, you can enter your SQL statement in an SQL worksheet and click the '**EXECUTE statement**' icon or press [F9]. The output displayed on the results tabbed page.



Retrieve Data From Specific Columns Of A Table

- You can retrieve data from specific columns of a table by specifying the column names, separated by commas.

SELECT column1, column2, column3 FROM table;

- Always check the columns of the tables before you write your sql queries to verify that you are typing the correct names for the columns.
- To do that you use the sqlplus DESCRIBE COMMAND OR DESC as shown below.

```
SQL> desc 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)

Retrieve Data From Specific Columns Of A Table

The example below will display all the employee IDs, first names and last names for all employees.

```
SQL> select employee_id, first_name, last_name  
2  from employees;
```

EMPLOYEE_ID	FIRST_NAME	LAST_NAME
100	Steven	King
101	Neena	Kochhar
102	Lex	De Haan
103	Alexander	Hunold
104	Bruce	Ernst
105	David	Austin
106	Valli	Pataballa
107	Diana	Lorentz
108	Nancy	Greenberg
109	Daniel	Faviet

Retrieve Data From Specific Columns Of A Table

You can also specify the columns in the order that you want them displayed in the output of the SELECT clause. As you can see from the screenshot below the output displays information for the employees in order of the last name.

```
SQL> select last_name, first_name, employee_id  
2  from employees;
```

LAST_NAME	FIRST_NAME	EMPLOYEE_ID
King	Steven	100
Kochhar	Neena	101
De Haan	Lex	102
Hunold	Alexander	103
Ernst	Bruce	104
Austin	David	105
Pataballa	Valli	106
Lorentz	Diana	107
Greenberg	Nancy	108

Remove Duplicate Rows From Tables

- The **DISTINCT** keyword is used to eliminate duplicate rows from the result set
- A query to retrieve the job types of the employees from the JOB_HISTORY table will display multiple rows with the same values because more than one employee works as an AC_Accountant or ST_CLERK.

```
SQL> select job_id from job_history;
```

```
JOB_ID
```

```
-----
```

```
AC_ACCOUNT
```

```
AC_ACCOUNT
```

```
AC_MGR
```

```
AD_ASST
```

```
IT_PROG
```

```
MK_REP
```

```
SA_MAN
```

```
SA_REP
```

```
ST_CLERK
```

```
ST CLERK
```

Remove Duplicate Rows From Tables

- In some situations, a unique set of rows is required and you can remove duplicate rows by using the DISTINCT keyword as shown below.

```
SQL> select distinct job_id from job_history;
```

```
JOB_ID
```

```
-----
```

```
AC_ACCOUNT
```

```
AC_MGR
```

```
AD_ASST
```

```
IT_PROG
```

```
MK_REP
```

```
SA_MAN
```

```
SA_REP
```

```
ST_CLERK
```

```
8 rows selected.
```

