Database Programming with PL/SQL

Review of SQL SELECT Statements

ORACLE ACADEMY

Objectives

This lesson covers the following objectives:

- Create a basic SQL statement including ORDER BY
- Perform and display arithmetic calculations
- Construct a query using a column alias
- Apply the concatenation operator
- Use literal values in a SELECT statement
- Use DISTINCT syntax in a query to eliminate duplicate rows
- Use conditional syntax including BETWEEN, IN, and LIKE, in a query

Purpose

PL/SQL is an extension to the SQL language. This means that the PL/SQL language builds on the SQL language.

Before diving into the complexities of PL/SQL, it is useful to have a strong foundation in SQL.

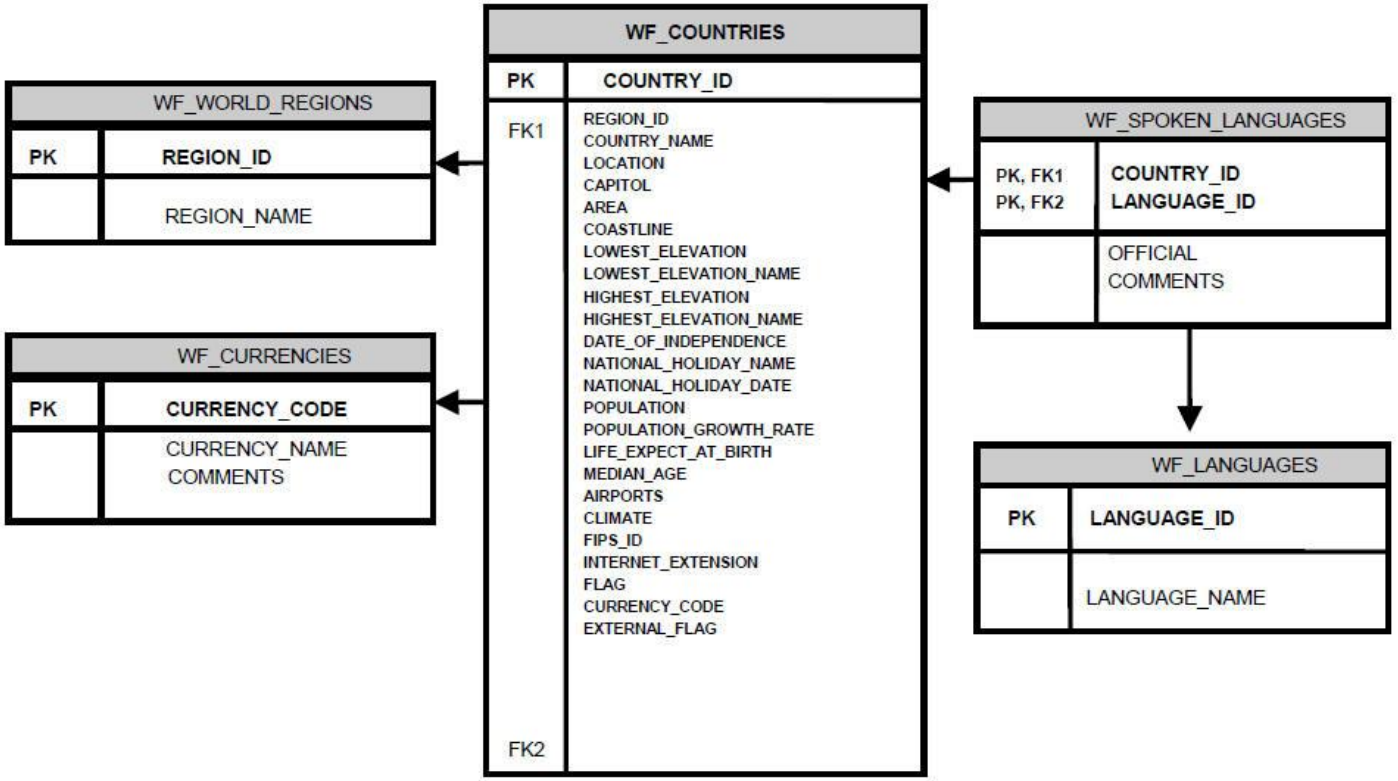
Selecting Data

`SELECT` is the keyword that retrieves columns from a table.

- The `FROM` clause specifies the *`tablename`*.
- `SELECT * FROM` *`tablename`* retrieves all the data in a table.
- `SELECT` *`<column list> FROM tablename`* retrieves the columns specified.
- The `WHERE` clause specifies a condition that restricts the rows returned by the `SELECT` statement.

Case Study

The review lessons reference the World Facts Schema.



Selecting Data

Use the SELECT statement to select data from the WF_COUNTRIES table.

```
SELECT country_name
    FROM wf_countries;
```

COUNTRY_NAME
Aruba
Antiqua and Barbuda
United Arab Emirates
Islamic Republic of Afghanistan
Peoples Democratic Republic of Algeria
Republic of Azerbaijan
Republic of Albania
Republic of Armenia
Principality of Andorra
Republic of Angola
More than 10 rows available. Increase rows selector to view more rows.

Sorting

ORDER BY specifies the display sequence of the result. The keywords ASC or DESC can be added after the column name to specify ascending or descending sequence.

```
SELECT country_name
   FROM wf_countries
   ORDER BY country_name;
```

COUNTRY_NAME
Anguilla
Antarctica
Antiqua and Barbuda
Arab Republic of Egypt
Argentine Republic
Aruba
Bailiwick of Guernsey
Bailiwick of Jersey
Barbados
Belize
More than 10 rows available. Increase rows selector to view more rows.

Calculations

The first example uses the multiplication operator to calculate the new area of Benin, if a land reclamation project increased its area by 2 percent.

```
SELECT country_name, area, area * 1.02
   FROM wf_countries
   WHERE country_id = 229;
```

COUNTRY_NAME	AREA	AREA*1.02
Republic of Benin	112620	114872.4

Column Aliases

The second example uses an alias to display the calculated value as "New Area".

```
SELECT country_name, area, area * 1.02 "New Area"
   FROM wf_countries
   WHERE country_id = 229;
```

COUNTRY_NAME	AREA	AREA*1.02
Republic of Benin	112620	114872.4

Concatenation

Concatenation means to connect or link together in a series. The concatenation operator is || (2 vertical bars sometimes referred to as “pipes”). A literal value is a character, a number, or a date that is included in the SELECT list and that is not a column name or a column alias. Literal values are often used with concatenation to create readable text output.

```
SELECT country_name || ' has an area of ' || area
     as "Readable Text"
FROM wf_countries;
```

Readable Text
Aruba has an area of 193
Antiqua and Barbuda has an area of 443
United Arab Emirates has an area of 82880
Islamic Republic of Afghanistan has an area of 647500
Peoples Democratic Republic of Algeria has an area of 2381740
Republic of Azerbaijan has an area of 86600
Republic of Albania has an area of 28748

DISTINCT

The DISTINCT keyword is used to eliminate duplicate rows from the output of an SQL statement. This example returns all the region IDs from the WF_COUNTRIES table.

```
SELECT region_id
   FROM wf_countries;
```

REGION_ID
29
29
145
34
15
145
39
145
39
18
More than 10 rows available. Increase rows selector to view more rows.

DISTINCT (cont.)

The DISTINCT keyword is used to eliminate duplicate rows from the output of an SQL statement. This example eliminates the duplicates.

```
SELECT DISTINCT region_id
   FROM wf_countries;
```

REGION_ID
34
151
30
29
155
13
11
21
14
5
More than 10 rows available. Increase rows selector to view more rows.

BETWEEN...AND

The BETWEEN...AND operator is used to select and display rows based on a range of values. The BETWEEN...AND condition is specified in the WHERE clause.

```
SELECT country_name, coastline
   FROM wf_countries
   WHERE coastline BETWEEN 500 AND 550;
```

COUNTRY_NAME	COASTLINE
Republic of Cote d Ivoire	515
Republic of Kenya	500
Republic of Latvia	531
Republic of Ghana	539
Paracel Islands	518
Commonwealth of Puerto Rico	501
Republic of Senegal	550

IN

The `IN` condition is used to test whether a value is in a specified set of values. The example shown selects countries that are in region 5 or 9.

```
SELECT region_id, country_name
   FROM wf_countries
   WHERE region_id IN (5,9);
```

REGION_ID	COUNTRY_NAME
9	Territory of American Somoa
5	Argentine Republic
9	Commonwealth of Australia
9	Antarctica
5	Republic of Bolivia
9	Solomon Islands
5	Federative Republic of Brazil
5	Republic of Chile
5	Republic of Colombia

LIKE

The `LIKE` condition allows you to select rows that match either literal strings or number patterns. The % and the underscore (_) are wildcard characters that you can use to construct a search string. The % symbol represents any sequence of zero or more characters. The underscore (_) symbol represents a single character.

```
SELECT country_name,
    national_holiday_name
  FROM wf_countries
  WHERE national_holiday_name
    LIKE '%Independence%';
```

COUNTRY_NAME	NATIONAL_HOLIDAY_NAME
Antiqua and Barbuda	Independence Day (National Day)
United Arab Emirates	Independence Day
Islamic Republic of Afghanistan	Independence Day
Republic of Albania	Independence Day
Republic of Armenia	Independence Day
Republic of Angola	Independence Day
Barbados	Independence Day
Republic of Botswana	Independence Day
Commonwealth of The Bahamas	Independence Day
Peoples Republic of Bangladesh	Independence Day
More than 10 rows available. Increase rows selector to view more rows.	

Terminology

Key terms used in this lesson included:

- `BETWEEN...AND`
- Concatenation
- `DISTINCT`
- `IN`
- `LIKE`

Summary

In this lesson, you should have learned how to:

- Create a basic SQL statement including ORDER BY
- Perform and display arithmetic calculations
- Construct a query using a column alias
- Apply the concatenation operator
- Use literal values in a SELECT statement
- Use DISTINCT syntax in a query to eliminate duplicate rows
- Use conditional syntax including BETWEEN, IN, and LIKE, in a query