



Autor: **Nicola Strappazon C.**  
e-mail: [nicola51980@gmail.com](mailto:nicola51980@gmail.com)  
Blog: <http://www.swapbytes.com/>  
Revisión: 17/11/11

# Lección 1

## Usando la sentencia SELECT

Este material se encuentra basado en el curso de Fundamentos a SQL de ORACLE, el cual es adaptado para el producto PostgreSQL, todos los ejemplos, códigos fuentes y la Base de Datos HR es propiedad de ORACLE.

# Objetivos

**Al completar esta lección usted podrá entender los siguientes puntos:**

- Ejecutar las sentencias básicas utilizando el comando **SELECT** de SQL.
- Podrá realizar operaciones aritméticas.
- Manejo básico de las cadenas de texto.

# Escribiendo Sentencias SQL

- SQL no es case-sensitive.
- SQL se puede escribir en una o mas líneas.
- Las palabras reservadas no se pueden abreviar.
- Las cláusulas se escriben en líneas separadas.
- Comentar para mejorar la comprensión.
- Toda palabra reservada se debe escribir en mayúscula para mejorar su comprensión.
- Toda sentencia debe terminar con el símbolo de (;) punto y coma.

# Convenciones

Símbolo	Significado
[ ]	Opcional.
	Un elemento de la lista.
[, ...]	Se repite.
{ }	Lista de elementos.

# Sintaxis Básica

- **SELECT** identifica las columnas que serán mostradas.
- **FROM** identifica la tabla donde provienen las columnas.

```
SELECT [ DISTINCT ] * | expresion [ AS alias ] [, ...]  
FROM   from_item [, ...];
```

# Seleccionando todo

```
SELECT *  
FROM departments;
```

department_name_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

■ ■ ■  
(27 rows)

# Seleccionando columnas

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

department_id	location_id
10	1700
20	1800
30	1700
40	2400
50	1500
60	1400
70	2700
80	2500
90	1700
100	1700

■ ■ ■  
(17 rows)

# Operadores aritméticos

Operador	Descripción	Precedencia
+	Adición	2
-	Sustracción	2
*	Multiplicación	1
/	División	1



# Usando los operadores aritméticos

1

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

last_name	salary
Russell	14000
...	

(107 rows)

2

```
SELECT last_name, salary + 100
FROM   employees;
```

last_name	?column?
Russell	14100
...	

(107 rows)

# Precedencia en los operadores

1

```
SELECT last_name, 10 * salary + 100
FROM employees;
```

last_name	?column?
Russell	140100
...	

(107 rows)

2

```
SELECT last_name, 10 * (salary + 100)
FROM employees;
```

last_name	?column?
Russell	141000
...	

(107 rows)

# Definiendo el valor NULL

- NULL es un valor inaplicable, no disponible, no asignado o desconocido.
- NULL no es un cero o un espacio en blanco.

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

last_name	job_id	salary	commission_pct
Taylor	SA_REP	8600	0.20
Livingston	SA_REP	8400	0.20
Johnson	SA_REP	6200	0.10
Kochhar	AD_VP	17000	
De Haan	AD_VP	17000	
Hunold	IT_PROG	9000	
■ ■ ■			
(107 rows)			

# Valores NULL en operaciones

**Cualquier operación aritmética que contenga valores nulos devolverán como resultado un valor nulo.**

```
SELECT last_name, salary, 10 * salary + commission_pct
FROM   employees;
```

last_name	salary	?column?
Taylor	8600	86000.20
Livingston	8400	84000.20
Johnson	6200	62000.10
Kochhar	17000	
De Haan	17000	
Hunold	9000	
■ ■ ■		

(107 rows)

# Definiendo un alias a las columnas

- Renombra el encabezado de una columna.
- Es muy útil cuando se realizan cálculos.
- El nuevo nombre se coloca después de la columna, no es opcional colocar la palabra reservada 'AS' entre la columna y el alias.
- Los nombres que contienen espacios, caracteres especiales o case-sensitive se colocan entre comillas dobles.

# Usando el alias

```
SELECT last_name AS name, salary, 10 * salary AS "New Salary"  
FROM employees;
```

name	salary	New Salary
Russell	14000	140000
Partners	13500	135000
Errazuriz	12000	120000
Cambrault	11000	110000
Zlotkey	10500	105000
Tucker	10000	100000
Bernstein	9500	95000
Hall	9000	90000
Olsen	8000	80000
Cambrault	7500	75000
■ ■ ■		

(107 rows)

# Concatenar columnas

- Permite unir varias cadena de texto o columnas.
- Es representada por dos barras verticales ( || ).
- La unión de varias cadenas de texto o columnas genera una nueva columna.

```
SELECT last_name || first_name AS "Employees"  
FROM   employees;
```

```
Employees  
-----  
RussellJohn  
PartnersKaren  
ErrazurizAlberto  
CambraultGerald  
■ ■ ■  
(107 rows)
```

# Caracteres literales

- El literal es un carácter, numero o fecha que están incluidas dentro de la sentencia SELECT.
- La fecha y el carácter literal son valores que se encierran en comillas simples (' ').
  - Por ejemplo:
    - 'U'
    - 'King'
    - '2007-12-01'



# Usando los caracteres literales

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

```
Employee Details  
-----  
Russell is a SA_MAN  
Partners is a SA_MAN  
Errazuriz is a SA_MAN  
Cambrault is a SA_MAN  
Zlotkey is a SA_MAN  
Tucker is a SA_REP  
Bernstein is a SA_REP  
Hall is a SA_REP  
Olsen is a SA_REP  
Cambrault is a SA_REP  
Tuvault is a SA_REP  
■ ■ ■  
(107 rows)
```

# Incluir caracteres especiales

- Utilizar cualquier delimitador.
- Incrementa la legibilidad y su uso.

```
SELECT last_name
       || E', it\'s assigned Manager ID: '
       || job_id AS "Employee Details"
FROM   employees;
```

Employee Details

```
-----
Russell, it's assigned Manager ID: SA_MAN
Partners, it's assigned Manager ID: SA_MAN
Errazuriz, it's assigned Manager ID: SA_MAN
Cambault, it's assigned Manager ID: SA_MAN
Zlotkey, it's assigned Manager ID: SA_MAN
Tucker, it's assigned Manager ID: SA_REP
```

```
■ ■ ■
```

```
(107 rows)
```

# Evitando registros duplicados

1

```
SELECT department_id
FROM   employees;
```

```
department_id
-----
              80
              80
. . .
(107 rows)
```

2

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

```
department_id
-----
              10
              20
. . .
(9 rows)
```

# Resumen

**En esta lección, usted debió entender como:**

- Usar la sentencia **SELECT** de SQL para:
  - Retornar todos los registros de una tabla.
  - Retornar columnas específicas de una tabla.
  - Definir un alias en las columnas.
  - Realizar operaciones aritméticas.
  - Manipular cadenas de texto y los literales.