



# Usando Sentencia SELECT para Recuperar Datos



- Cómo utilizar sentencia SELECT básica para mostrar datos desde las Tablas de la Base de Datos.
- Cómo utilizar operadores matemáticos en una sentencia SELECT.
- Cómo unir valores/columnas/expresiones en una sentencia SELECT.
- Cómo utilizar Alias para asignar nombres lógicos a las columnas y expresiones obtenidas en una sentencia SELECT.
- Cómo mostrar la información en un orden específico.





# SENTENCIA SELECT

# Capacidades de la Sentencia SELECT

## PROYECCIÓN


Tabla 1

## SELECCIÓN


Tabla 1


Tabla 1

JOIN


Tabla 2

# Seleccionando las Columnas de una Tabla

**SELECT** permite  
mostrar una o más  
columnas de las  
tablas, además de  
expresiones

Si se desea mostrar  
más de una  
columna o  
expresión, éstas se  
deben separar con  
comas

- Sintaxis:

```
SELECT * | { [ DISTINCT ] columna | expresión [alias], ... }  
FROM tabla  
[WHERE condición]  
[ORDER BY {columna, alias, expresión, posición_numérica} [ASC|DESC]];
```

# Seleccionando Todas las Columnas de la Tabla

- Ejemplo:

```
SELECT *  
FROM departments;
```

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	30	Purchasing	114	1700
4	40	Human Resources	203	2400
5	50	Shipping	121	1500
6	60	IT	103	1400
7	70	Public Relations	204	2700
8	80	Sales	145	2500
9	90	Executive	100	1700
10	100	Finance	108	1700
11	110	Accounting	205	1700
.....				
23	230	IT Helpdesk	(null)	1700
24	240	Government Sales	(null)	1700
25	250	Retail Sales	(null)	1700
26	260	Recruiting	(null)	1700
27	270	Payroll	(null)	1700

# Seleccionando Columnas Específicas de la Tabla

- Ejemplo:

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

	DEPARTMENT_ID	LOCATION_ID
1	10	1700
2	20	1800
3	30	1700
4	40	2400
5	50	1500
6	60	1400
7	70	2700
8	80	2500
9	90	1700
10	100	1700
11	110	1700
.....		
24	240	1700
25	250	1700
26	260	1700
27	270	1700

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

	LOCATION_ID	DEPARTMENT_ID
1	1700	10
2	1800	20
3	1700	30
4	2400	40
5	1500	50
6	1400	60
7	2700	70
8	2500	80
9	1700	90
10	1700	100
11	1700	110
.....		
24	1700	240
25	1700	250
26	1700	260
27	1700	270



# Seleccionando Columnas Específicas de la Tabla

- Ejemplo:

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

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY
1	100	Steven	King	24000
2	101	Neena	Kochhar	17000
3	102	Lex	De Haan	17000
4	103	Alexander	Hunold	9000
5	104	Bruce	Ernst	6000
6	105	David	Austin	4800
7	106	Valli	Pataballa	4800
8	107	Diana	Lorentz	4200
9	108	Nancy	Greenberg	12008
10	109	Daniel	Faviet	9000
.....				
.....				
104	203	Susan	Mavris	6500
105	204	Hermann	Baer	10000
106	205	Shelley	Higgins	12008
107	206	William	Gietz	8300



# Operadores Matemáticos

Para efectuar cálculos con los datos de las tablas se deben usar expresiones aritméticas

Una expresión aritmética puede contener nombre de columnas, constantes de valores numéricos y operadores aritméticos

Los operadores aritméticos se pueden usar en cualquier cláusula de una sentencia DML excepto en la cláusula FROM

- Los operadores que se pueden utilizar en una sentencia SQL son:

OPERADOR	DESCRIPCIÓN
+	Suma
-	Resta
*	Multiplicación
/	División

# Usando Operadores Aritméticos

En los operadores con igual prioridad se ejecutan desde izquierda a derecha

La multiplicación y la división se ejecutan antes que la suma y la resta

Se puede evitar cumplir la prioridad de los operadores usando paréntesis; lo que está en paréntesis es lo que se ejecuta primero

Cuando una expresión aritmética se calcula sobre valores NULOS (NULL) el resultado de la expresión es siempre NULO (NULL)

- Ejemplo:

```
SELECT last_name, salary, salary + 300  
FROM employees;
```

	LAST_NAME	SALARY	SALARY+300
1	King	24000	24300
2	Kochhar	17000	17300
3	De Haan	17000	17300
4	Hunold	9000	9300
5	Ernst	6000	6300
6	Austin	4800	5100
7	Pataballa	4800	5100
8	Lorentz	4200	4500
9	Greenberg	12008	12308
10	Faviet	9000	9300
.....			
104	Mavris	6500	6800
105	Baer	10000	10300
106	Higgins	12008	12308
107	Gietz	8300	8600

- Ejemplo:

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

	LAST_NAME	SALARY	12*SALARY+100
1	King	24000	288100
2	Kochhar	17000	204100
3	De Haan	17000	204100
4	Hunold	9000	108100
5	Ernst	6000	72100
6	Austin	4800	57700
7	Pataballa	4800	57700
8	Lorentz	4200	50500
9	Greenberg	12008	144196
10	Faviet	9000	108100
.....			
104	Mavris	6500	78100
105	Baer	10000	120100
106	Higgins	12008	144196
107	Gietz	8300	99700



- Ejemplo:

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

	LAST_NAME	SALARY	COMMISSION_PCT	12*SALARY*COMMISSION_PCT
1	King	24000	(null)	(null)
2	Kochhar	17000	(null)	(null)
3	De Haan	17000	(null)	(null)
4	Hunold	9000	(null)	(null)
5	Ernst	6000	(null)	(null)
6	Austin	4800	(null)	(null)
7	Pataballa	4800	(null)	(null)
.....				
64	Greene	9500	0,15	17100
65	Marvins	7200	0,1	8640
66	Lee	6800	0,1	8160
67	Ande	6400	0,1	7680
68	Banda	6200	0,1	7440
69	Ozer	11500	0,25	34500
.....				
104	Mavris	6500	(null)	(null)
105	Baer	10000	(null)	(null)
106	Higgins	12008	(null)	(null)
107	Gietz	8300	(null)	(null)



# Alias de Columnas, Concatenación y Valores Duplicados

## Renombra una columna

Debe ir entre doble comillas si posee espacios, caracteres especiales o es case-sensitive

Se muestran en lugar del nombre real de la columna o expresión

Va a continuación del nombre de columna o expresión

Útil cuando se efectúan cálculos

# Definiendo Alias de Columnas

- Ejemplo:

```
SELECT last_name AS apellido , salary salario  
FROM employees;
```

	APELLIDO	SALARIO
1	King	24000
2	Kochhar	17000
3	De Haan	17000
4	Hunold	9000
5	Ernst	6000
6	Austin	4800
7	Pataballa	4800
8	Lorentz	4200
9	Greenberg	12008
10	Faviet	9000
.....		
104	Mavris	6500
105	Baer	10000
106	Higgins	12008
107	Gietz	8300

# Definiendo Alias de Columnas

- Ejemplo:

```
SELECT last_name, salary "Salario" , salary* 1.25 "Salario Aumentado en 25%"  
FROM employees;
```

	LAST_NAME	Salario	Salario Aumentado en 25%
1	King	24000	30000
2	Kochhar	17000	21250
3	De Haan	17000	21250
4	Hunold	9000	11250
5	Ernst	6000	7500
6	Austin	4800	6000
7	Pataballa	4800	6000
8	Lorentz	4200	5250
9	Greenberg	12008	15010
10	Faviet	9000	11250
11	Chen	8200	10250
.....			
105	Baer	10000	12500
106	Higgins	12008	15010
107	Gietz	8300	10375

**Está representado por dos  
barras verticales (||)**

**Permite unir columnas y/o  
cadena de caracteres literales y  
formas una sola columna de  
salida**

**Si se desean concatenar fechas  
y/o caracteres literales, estos  
deben ir entre comillas simples**

**Crea una columna resultante  
que es una expresión de  
caracteres**

**Si a una columna se concatena  
un NULO, el resultado es una  
expresión de tipo caracter**



- Ejemplo:

```
SELECT employee_id "IDENTIFICACION EMPLEADO",  
       first_name || ' ' || last_name "NOMBRE DEL EMPLEADO",  
       salary salario  
FROM employees;
```

	IDENTIFICACION EMPLEADO	NOMBRE DEL EMPLEADO	SALARIO
1	100	Steven King	24000
2	101	Neena Kochhar	17000
3	102	Lex De Haan	17000
4	103	Alexander Hunold	9000
5	104	Bruce Ernst	6000
6	105	David Austin	4800
7	106	Valli Pataballa	4800
8	107	Diana Lorentz	4200
9	108	Nancy Greenberg	12008
10	109	Daniel Faviet	9000
.....			
104	203	Susan Mavris	6500
105	204	Hermann Baer	10000
106	205	Shelley Higgins	12008
107	206	William Gietz	8300

- Ejemplo:

```
SELECT last_name || ' pertenece al departamento ' ||  
       department_id "Detalle de Empleados"  
FROM employees;
```

Detalle de Empleados	
1	King pertenece al departamento 90
2	Kochhar pertenece al departamento 90
3	De Haan pertenece al departamento 90
4	Hunold pertenece al departamento 60
5	Ernst pertenece al departamento 60
6	Austin pertenece al departamento 60
7	Pataballa pertenece al departamento 60
8	Lorentz pertenece al departamento 60
9	Greenberg pertenece al departamento 100
10	Faviet pertenece al departamento 100
.....	
104	Mavris pertenece al departamento 40
105	Baer pertenece al departamento 70
106	Higgins pertenece al departamento 110
107	Gietz pertenece al departamento 110

- Ejemplo:

```
SELECT 'El salario mensual del empleado ' || last_name || ' es de $ ' || salary salarios  
FROM employees;
```

	⚡ SALARIOS
1	El salario mensual del empleado King es de \$ 24000
2	El salario mensual del empleado Kochhar es de \$ 17000
3	El salario mensual del empleado De Haan es de \$ 17000
4	El salario mensual del empleado Hunold es de \$ 9000
5	El salario mensual del empleado Ernst es de \$ 6000
6	El salario mensual del empleado Austin es de \$ 4800
7	El salario mensual del empleado Pataballa es de \$ 4800
8	El salario mensual del empleado Lorentz es de \$ 4200
9	El salario mensual del empleado Greenberg es de \$ 12008
10	El salario mensual del empleado Faviet es de \$ 9000
.....	
104	El salario mensual del empleado Mavris es de \$ 6500
105	El salario mensual del empleado Baer es de \$ 10000
106	El salario mensual del empleado Higgins es de \$ 12008
107	El salario mensual del empleado Gietz es de \$ 8300

# Valores Duplicados en las Columnas

Para mostrar sólo las filas diferentes se debe usar la palabra **DISTINCT** en la cláusula **SELECT**

Al consultar columnas que no son parte de la PK de la tabla, se muestran todas las filas (incluidas las duplicadas)

Al usar **DISTINCT**, se pueden incluir todas las columnas de la tabla que se requieran visualizar

Al usar **DISTINCT** afecta a todas las columnas que se seleccionan

# Valores Duplicados en las Columnas

- Ejemplo:

```
SELECT department_id  
FROM employees;
```

	DEPARTMENT_ID
1	90
2	90
3	90
4	60
5	60
6	60
7	60
8	60
9	100
10	100
.....	
104	40
105	70
106	110
107	110

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

	DEPARTMENT_ID
1	50
2	40
3	110
4	90
5	30
6	70
7	(null)
8	10
9	20
10	60
11	100
12	80

# Cláusula ORDER BY



# Ordenando las Filas Recuperadas

Las filas que retorna una query no tienen un orden definido. Para ordenarlas se debe usar **ORDER BY**

La cláusula **ORDER BY** va al final de la sentencia **SELECT** y permite ordenar en forma Ascendente (defecto) o Descendente

Se puede especificar el nombre de la columna, un alias, una expresión o la posición de la columna

- Sintaxis:

```
SELECT * | { [ DISTINCT ] columna | expresión [alias], ... }  
FROM tabla  
[WHERE condición]  
[ORDER BY {columna, alias, expresión, posición_numérica} [ASC|DESC]];
```

# Ordenando las Filas Recuperadas

- Ejemplo:

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

	LAST_NAME	JOB_ID	HIRE_DATE
1	De Haan	AD_VP	13/01/2001
2	Gietz	AC_ACCOUNT	07/06/2002
3	Baer	PR_REP	07/06/2002
4	Mavris	HR_REP	07/06/2002
5	Higgins	AC_MGR	07/06/2002
6	Faviet	FI_ACCOUNT	16/08/2002
7	Greenberg	FI_MGR	17/08/2002
8	Raphaely	PU_MAN	07/12/2002
9	Kaufling	ST_MAN	01/05/2003
10	Khoo	PU_CLERK	18/05/2003

104	Markle	ST_CLERK	08/03/2008
105	Ande	SA_REP	24/03/2008
106	Banda	SA_REP	21/04/2008
107	Kumar	SA_REP	21/04/2008

```
SELECT last_name, job_id, hire_date
FROM employees
ORDER BY hire_date DESC;
```

	LAST_NAME	JOB_ID	HIRE_DATE
1	Kumar	SA_REP	21/04/2008
2	Banda	SA_REP	21/04/2008
3	Ande	SA_REP	24/03/2008
4	Markle	ST_CLERK	08/03/2008
5	Lee	SA_REP	23/02/2008
6	Philtanker	ST_CLERK	06/02/2008
7	Geoni	SH_CLERK	03/02/2008
8	Zlotkey	SA_MAN	29/01/2008
9	Marvins	SA_REP	24/01/2008
10	Grant	SH_CLERK	13/01/2008

104	Mavris	HR_REP	07/06/2002
105	Baer	PR_REP	07/06/2002
106	Higgins	AC_MGR	07/06/2002
107	De Haan	AD_VP	13/01/2001

# Ordenando las Filas Recuperadas

- Ejemplo:

```
SELECT employee_id, last_name, salary,  
       salary*12 "Salario Anual"  
FROM employees  
ORDER BY "Salario Anual" ;
```

```
SELECT employee_id, last_name, salary,  
       salary*12 "Salario Anual"  
FROM employees  
ORDER BY salary*12;
```

	EMPLOYEE_ID	LAST_NAME	SALARY	Salario Anual
1	132	Olson	2100	25200
2	128	Markle	2200	26400
3	136	Philtanker	2200	26400
4	135	Gee	2400	28800
5	127	Landry	2400	28800
6	119	Colmenares	2500	30000
7	131	Marlow	2500	30000
8	140	Patel	2500	30000
9	144	Vargas	2500	30000
10	182	Sullivan	2500	30000

104	145	Russell	14000	168000
105	102	De Haan	17000	204000
106	101	Kochhar	17000	204000
107	100	King	24000	288000

# Ordenando las Filas Recuperadas

- Ejemplo:

```
SELECT employee_id, last_name, salary,  
       salary*12 "Salario Anual"  
FROM employees  
ORDER BY "SALARIO ANUAL" ;
```

ORA-00904: "SALARIO ANUAL": identificador no válido  
00904. 00000 - "%s: invalid identifier"  
\*Cause:  
\*Action:  
Error en la línea: 3, columna: 10

- Ejemplo:

```
SELECT last_name, department_id, salary  
FROM employees  
ORDER BY department_id, 3 DESC;
```

	LAST_NAME	DEPARTMENT_ID	SALARY
1	Whalen	10	4400
2	Hartstein	20	13000
3	Fay	20	6000
4	Raphaely	30	11000
5	Khoo	30	3100
6	Baida	30	2900
7	Tobias	30	2800
8	Himuro	30	2600
9	Colmenares	30	2500
10	Mavris	40	6500

104	Popp	100	6900
105	Higgins	110	12008
106	Gietz	110	8300
107	Grant	(null)	7000

# Ordenando las Filas Recuperadas

- Ejemplo:

```
SELECT department_id, job_id, last_name, salary
FROM employees
ORDER BY department_id, job_id, last_name;
```

	DEPARTMENT_ID	JOB_ID	LAST_NAME	SALARY
1	10	AD_ASST	Whalen	4400
2	20	MK_MAN	Hartstein	13000
3	20	MK_REP	Fay	6000
4	30	PU_CLERK	Baida	2900
5	30	PU_CLERK	Colmenares	2500
6	30	PU_CLERK	Himuro	2600
7	30	PU_CLERK	Khoo	3100
8	30	PU_CLERK	Tobias	2800
9	30	PU_MAN	Raphaely	11000
10	40	HR_REP	Mavris	6500
.....				
99	100	FI_ACCOUNT	Chen	8200
100	100	FI_ACCOUNT	Faviet	9000
101	100	FI_ACCOUNT	Popp	6900
102	100	FI_ACCOUNT	Sciarra	7700
103	100	FI_ACCOUNT	Urman	7800
104	100	FI_MGR	Greenberg	12008
105	110	AC_ACCOUNT	Gietz	8300
106	110	AC_MGR	Higgins	12008
107	(null)	SA_REP	Grant	7000

# Ordenando las Filas Recuperadas

- Ejemplo:

```
SELECT employee_id "ID EMPLEADO",  
       salary salario ,  
       commission_pct "PORCENTAJE COMISION"  
FROM employees  
ORDER BY commission_pct, salary;
```

```
SELECT employee_id "ID EMPLEADO",  
       salary salario ,  
       commission_pct "PORCENTAJE COMISION"  
FROM employees  
ORDER BY commission_pct NULLS LAST, salary;
```

	ID EMPLEADO	SALARIO	PORCENTAJE COMISION
1	173	6100	0,1
2	179	6200	0,1
3	167	6200	0,1
4	166	6400	0,1
5	165	6800	0,1
6	164	7200	0,1
7	178	7000	0,15
8	155	7000	0,15
9	172	7300	0,15
10	171	7400	0,15
11	163	9500	0,15
12	154	7500	0,2
13	153	8000	0,2
14	177	8400	0,2
15	176	8600	0,2
16	170	9600	0,2
17	169	10000	0,2
18	149	10500	0,2

100	204	10000	(null)
101	114	11000	(null)
102	205	12008	(null)
103	108	12008	(null)
104	201	13000	(null)
105	101	17000	(null)
106	102	17000	(null)
107	100	24000	(null)



- Se explicó cómo utilizar sentencia SELECT básica para mostrar datos desde las Tablas de la Base de Datos.
- Se explicó cómo utilizar operadores matemáticos en una sentencia SELECT.
- Se explicó cómo unir valores/columnas/expresiones en una sentencia SELECT.
- Se explicó cómo utilizar Alias para asignar nombres lógicos a las columnas y expresiones obtenidas en una sentencia SELECT.
- Se explicó cómo mostrar la información en un orden específico.