

BOOTCAMP ORACLE Y JAVA
MODULO II: ORACLE
Presentado por: Herbert Fernández Tamayo

DESARROLLO DE PRACTICAS SQL

1 MODULO 4

1.1 Practicas con Select

- Visualizar el nombre y el número de teléfono de los empleados

```
oracle > practicas > s04 > s04.sql > ...
1   select first_name,
2   |   phone_number
3   from employees;|
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY

Fetched 107 rows in 0.067 seconds

	FIRST_NAME	PHONE_NUMBER
1	Steven	515.123.4567
2	Neena	515.123.4568
3	Lex	515.123.4569
4	Alexander	590.423.4567
5	Bruce	590.423.4568
6	David	590.423.4569
7	Valli	590.423.4560
8	Diana	590.423.5567
9	Nancy	515.124.4569
10	Daniel	515.124.4169
11	John	515.124.4269
12	Tonya	515.124.4269

- Visualizar el nombre y el tipo de trabajo de los empleados (FIRST_NAME, JOB_ID). Debe aparecer en la cabecera NOMBRE Y Tipo de Trabajo.

```
4
5 select first_name as nombre,
6      ... job_id as "Tipo de Trabajo"
7  from employees;
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR C

atched 107 rows in 0.051 seconds

	NOMBRE	Tipo de Trabajo
1	Ellen	SA_REP
2	Sundar	SA_REP
3	Mozhe	ST_CLERK
4	David	IT_PROG
5	Hermann	PR_REP
6	Shelli	PU_CLERK
7	Amit	SA_REP
8	Elizabeth	SA_REP

- Selecciona todas las columnas de la tabla REGIONS

```
8
9 select *
10   from regions;
11
12 select country_name
13   from countries;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR

rows fetched: 4 in 0.034 seconds

	REGION_ID	REGION_NAME
1	1	Europe
2	2	Americas
3	3	Asia
4	4	Middle East and Africa

- Indicar los nombres de los países de la tabla COUNTRIES

```
10     |     from regions,  
11     |  
12     | select country_name  
13     |     |     from countries;  
14     |  
15     | select street_address as "Dirección",  
16     |     |     |     |     |     |  
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS    C
```

All rows fetched: 25 in 0.040 seconds

	COUNTRY_NAME
1	Argentina
2	Australia
3	Belgium
4	Brazil
5	Canada
6	Switzerland
7	China
8	Germany

- Seleccionar las columnas STREET_ADDRESS, CITY, STATE_PROVINCE de la table LOCATIONS. Debemos poner las columnas como dirección, Ciudad y Estado

```

4
5 select street_address as "Dirección",
6      ... city as "Ciudad",
7      ... state_province as "Estado"
8   from locations;

```

rows fetched: 23 in 0.020 seconds

	Dirección	Ciudad	Estado
1	1297 Via Cola di Rie	Roma	(null)
2	93091 Calle della Testa	Venice	(null)
3	2017 Shinjuku-ku	Tokyo	Tokyo Prefecture
4	9450 Kamiya-cho	Hiroshima	(null)
5	2014 Jabberwocky Rd	Southlake	Texas
6	2011 Interiors Blvd	South San Francisco	California
7	2007 Zagora St	South Brunswick	New Jersey
8	2004 Charade Rd	Seattle	Washington

1.2 Practicas con Literales

Crear la consulta para visualizar los siguientes datos, usando el operador de concatenación ||

Crear la siguiente consulta

```

9
10 SELECT 'El empleado '||first_name||' del departamento '||department_id||' tiene un salario de '
11      ... ||salary AS DATOS from employees;
12
13 select 'La calle '||street_address||' pertenece a la ciudad:'||city as "calle y ciudad" from
14 locations;

```

atched 107 rows in 0.057 seconds

	DATOS
1	El empleado Steven del departamento 90 tiene un salario de 24000
2	El empleado Neena del departamento 90 tiene un salario de 17000
3	El empleado Lex del departamento 90 tiene un salario de 17000
4	El empleado Alexander del departamento 60 tiene un salario de 9000
5	El empleado Bruce del departamento 60 tiene un salario de 6000
6	El empleado David del departamento 60 tiene un salario de 4800
7	El empleado Valli del departamento 60 tiene un salario de 4800
8	El empleado Diana del departamento 60 tiene un salario de 4200

```
25 ✨
26 select 'La calle '||street_address||' pertenece a la ciudad:'||city as "Calle y Ciudad" from
27 locations;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR COMMENTS

All rows fetched: 23 in 0.014 seconds

Calle y Ciudad
La calle 1297 Via Cola di Rie pertenece a la ciudad:Roma
La calle 93091 Calle della Testa pertenece a la ciudad:Venice
La calle 2017 Shinjuku-ku pertenece a la ciudad:Tokyo
La calle 9450 Kamiya-cho pertenece a la ciudad:Hirosshima
La calle 2014 Jabberwocky Rd pertenece a la ciudad:Southlake
La calle 2011 Interiors Blvd pertenece a la ciudad:South San Francisco
La calle 2007 Zagora St pertenece a la ciudad:South Brunswick
La calle 2004 Charade Rd pertenece a la ciudad:Seattle

1.3 Practicas con Distinct

- Visualizar las ciudades donde hay departamentos, de la tabla locations. No deben salir repetidos

25

26

27

```
25 ✨
26 SELECT DISTINCT CITY FROM LOCATIONS; Ctrl+I to co
27
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RE

All rows fetched: 23 in 0.051 seconds

	CITY
1	Beijing
2	Bern
3	Bombay
4	Geneva
5	Hiroshima
6	London
7	Mexico City
8	Munich
9	Oxford

- Visualizar los distintos tipos de JOB_ID por departamento de la tabla Employees.

```

7
8 SELECT DISTINCT DEPARTMENT_ID, JOB_ID FROM EMPLOYEES;

```

rows fetched: 20 in 0.090 seconds

	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

2 MODULO 5

2.1 Practicas con Where

- Averigua los empleados que trabajen en el departamento 100

```

acre > practicas > s05 > s05.sql > ...
1 select *
2 from employees
3 where department_id = 100;
4
5 select street_address,

```

rows fetched: 6 in 0.081 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE
1	108	Nancy	Greenberg	NGREENBE	515.124.4569	17-APR-05
2	109	Daniel	Faviet	DFAVIET	515.124.4169	16-JAN-06
3	110	John	Chen	JCHEN	515.124.4269	28-NOV-05
4	111	Ismael	Sciarra	ISCIARRA	515.124.4369	30-MAR-06
5	112	Jose Manuel	Urman	JMURMAN	515.124.4469	07-JUN-06
6	113	Luis	Popp	LPOPP	515.124.4567	07-JUN-06

- Usando la tabla LOCATIONS, averigua el nombre de la Ciudad (city) y la dirección (Street_address) de los departamentos situados en Estados Unidos (COUNTRY_ID=US)

```

5  select street_address,
6  ...|... city
7  ...from locations
8  ...where country_id = 'US';
9

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

All rows fetched: 4 in 0.039 seconds

	STREET_ADDRESS	CITY
1	2014 Jabberwocky Rd	Southlake
2	2011 Interiors Blvd	South San Francisco
3	2007 Zagora St	South Brunswick
4	2004 Charade Rd	Seattle

- Visualiza los países que están en la región 3. (REGION_ID de la tabla COUNTRIES

```

oracle > practicas > s05 > s05.sql > ...
10  select *
11  ...from countries
12  ...where region_id = 3;
13
14  select first_name,

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

All rows fetched: 6 in 0.021 seconds

	COUNTRY_ID	COUNTRY_NAME	REGION_ID
1	AU	Australia	3
2	CN	China	3
3	IN	India	3
4	JP	Japan	3
5	ML	Malaysia	3
6	SG	Singapore	3

- Averiguar el nombre y salario de los empleados que NO tengan como jefe al MANAGER 114 (columna MANAGER_ID)

oracle > practicas > s05 > s05.sql > ...

```

14 select first_name,
15      salary
16   from employees
17 where manager_id <> 114;
18

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

Fetched 101 rows in 0.040 seconds

	FIRST_NAME	SALARY
1	Neena	17000
2	Lex	17000
3	Alexander	9000
4	Bruce	6000
5	David	4800
6	Valli	4800
7	Diana	4200
8	Nancy	12008
9	Daniel	9000

- Visualizar los empleados que empezaron a trabajar a partir del año 2006

oracle > practicas > s05 > s05.sql > ...

```

19 select *
20   from employees
21  where hire_date >= '01-01-06';
22
23 select *
24   from employees

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

All rows fetched: 54 in 0.034 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
1	103	Alexander	Hunold	AHUNOLD	590.423.4567
2	104	Bruce	Ernst	BERNST	590.423.4568
3	106	Valli	Pataballa	VPATABAL	590.423.4560
4	107	Diana	Lorentz	DLORENTZ	590.423.5567
5	112	Jose Manuel	Urman	JMURMAN	515.124.4469
6	113	Luis	Popp	LPOPP	515.124.4567
7	118	Guy	Himuro	GHIMURO	515.127.4565
8	119	Karen	Colmenares	KCOLMENA	515.127.4566
9	124	Kevin	Mourgos	KMOURGOS	650.123.5234

- Seleccionar los empleados que tenga como tipo de trabajo ‘ST_CLERK’



```

23 select *
24   from employees
25  where job_id = 'ST_CLERK';
26
27 select *

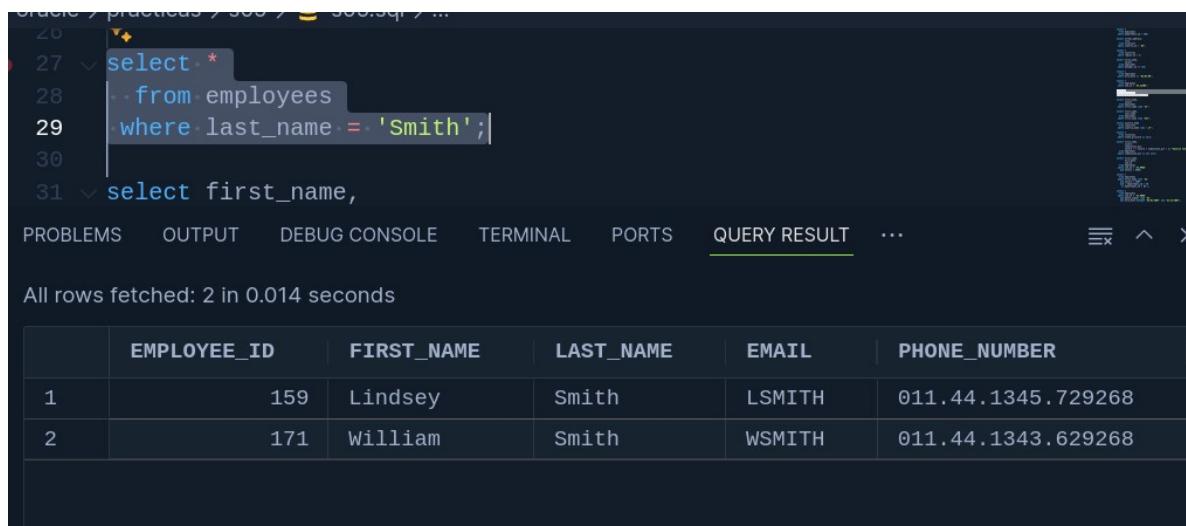
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

All rows fetched: 20 in 0.015 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
1	125	Julia	Nayer	JNAYER	650.124.1214
2	126	Irene	Mikkilineni	IMIKKILI	650.124.1224
3	127	James	Landry	JLANDRY	650.124.1334
4	128	Steven	Markle	SMARKLE	650.124.1434
5	129	Laura	Bissot	LBISSOT	650.124.5234
6	130	Mozhe	Atkinson	MATKINSO	650.124.6234
7	131	James	Marlow	JAMRLOW	650.124.7234
8	132	TJ	Olson	TJOLSON	650.124.8234

- Indicar los datos de los empleados que tengan como apellidos “Smith” (LAST_NAME)



```

26
27 select *
28   from employees
29  where last_name = 'Smith';
30
31 select first_name,

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

All rows fetched: 2 in 0.014 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
1	159	Lindsey	Smith	LSMITH	011.44.1345.729268
2	171	William	Smith	WSMITH	011.44.1343.629268

2.2 Practicas con Like

- Indicar los datos de los empleados cuyo FIRST_NAME empieza por 'J'

The screenshot shows a code editor interface with a dark theme. On the left, there is a vertical file tree. The main area contains the following SQL code:

```
30
31 select first_name,
32      ... salary
33   from employees
34  where first_name like 'J%';
35
```

Below the code, there are tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, QUERY RESULT, and a menu icon. The QUERY RESULT tab is selected, indicated by a green underline. The output below the tabs shows:

All rows fetched: 16 in 0.013 seconds

	FIRST_NAME	SALARY
1	John	8200
2	Julia	3400
3	Jennifer	3600
4	Jean	3100
5	Janette	10000
6	James	2400
7	Jack	8400
8	Jason	3300

- Averiguar los empleados que comienzan por 'S' y terminan en 'n'

The screenshot shows a code editor interface with a dark theme. On the left, there is a vertical file tree. The main area contains the following SQL code:

```
oracle > practicas > s05 > s05.sql > ...
36 select first_name,
37      ... last_name
38   from employees
39  where first_name like 'S%n';
40
```

Below the code, there are tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, QUERY RESULT, and a menu icon. The QUERY RESULT tab is selected, indicated by a green underline. The output below the tabs shows:

All rows fetched: 4 in 0.014 seconds

	FIRST_NAME	LAST_NAME
1	Steven	King
2	Steven	Markle
3	Susan	Mavris
4	Stephen	Stiles

- Indicar los países que tienen una “r” en la segunda letra (Tabla COUNTRIES)

```
oracle > practicas > s05 > s05.sql > ...
41  select country_name
42  from countries
43  where country_name like '_r%';
44
45  select *
46  from locations
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
All rows fetched: 3 in 0.062 seconds
```

	COUNTRY_NAME
1	Argentina
2	Brazil
3	France

2.3 Practicas con Is Null

- Listar las ciudades de la tabla LOCATIONS no tienen STATE_PROVINCE

```

45 select *
46   from locations
47  where state_province is null;
48
49 select first_name,

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

All rows fetched: 6 in 0.013 seconds

	LOCATION_ID	STREET_ADDRESS	POSTAL_CODE	CITY	STATE_PROVINCE
1	1000	1297 Via Cola di Rie	00989	Roma	(null)
2	1100	93091 Calle della Testa	10934	Venice	(null)
3	1300	9450 Kamiya-cho	6823	Hiroshima	(null)
4	2000	40-5-12 Laogianggen	190518	Beijing	(null)
5	2300	198 Clementi North	540198	Singapore	(null)
6	2400	8204 Arthur St	(null)	London	(null)

- Averiguar el nombre, salario y comisión de aquellos empleados que tienen comisión. También debemos visualizar una columna calculada denominada “Sueldo Total”, que sea el sueldo más la comisión

```

49 select first_name,
50      salary,
51      commission_pct,
52      salary + (salary * commission_pct) as "Salario total"
53  from employees
54 where commission_pct is not null;
55
56 select first_name

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

All rows fetched: 35 in 0.011 seconds

	FIRST_NAME	SALARY	COMMISSION_PCT	Salario total
1	John	14000	0.4	19600
2	Karen	13500	0.3	17550
3	Alberto	12000	0.3	15600
4	Gerald	11000	0.3	14300
5	Eleni	10500	0.2	12600
6	Peter	10000	0.3	13000
7	David	9500	0.25	11875

2.4 Practicas con AND, OR, NOT

- Obtener el nombre y la fecha de la entrada y el tipo de trabajo de los empleados que sean IT_PROG y que ganen menos de 6000 dólares

```
55
56 select first_name,
57      ... hire_date,
58      ... job_id
59   from employees
60  where job_id = 'IT_PROG'
61    and salary < 6000;
62
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

All rows fetched: 3 in 0.050 seconds

	FIRST_NAME	HIRE_DATE	JOB_ID
1	David	25/06/05	IT_PROG
2	Valli	05/02/06	IT_PROG
3	Diana	07/02/07	IT_PROG

- Seleccionar los empleados que trabajen en el departamento 50 o 80, cuyo nombre comience por S y que ganen más de 3000 dólares.

```
62
63 select *
64   from employees
65  where first_name like 'S%'
66    and salary > 3000
67    and ( department_id = 50
68      or department_id = 80 );
69
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

All rows fetched: 7 in 0.012 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
1	166	Sundar	Ande	SANDE	011.44.1346.629268
2	192	Sarah	Bell	SBELL	650.501.1876
3	173	Sundita	Kumar	SKUMAR	011.44.1343.329268
4	194	Samuel	McCain	SMCCAIN	650.501.3876
5	161	Sarath	Sewall	SSEWALL	011.44.1345.529268
6	138	Stephen	Stiles	SSTILES	650.121.2034
7	123	Shanta	Vollman	SVOLLMAN	650.123.4234

- ¿Qué empleados de job_id IT_PROG tienen un prefijo 5 en el teléfono y entraron en la empresa en el año 2007?

```
oracle > practicas > s05 > s05.sql > ...
09
70 select *
71   from employees
72  where job_id = 'IT_PROG'
73    and phone_number like '5%'
74    and hire_date between '01-01-2007' and '31-12-2007';
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

All rows fetched: 2 in 0.013 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE
1	104	Bruce	Ernst	BERNST	590.423.4568	21/05/2007
2	107	Diana	Lorentz	DLORENTZ	590.423.5567	07/05/2007

3 MODULO 6

3.1 Practicas con funciones carácter

- En la tabla LOCATIONS, averiguar las ciudades que son de Canada o Estados Unidos (Country_id=CA o US) y que la longitud del nombre de la calle sea superior a 15.

```
oracle > practicas > s06 > s06.sql > ...
1 select street_address,
2       city
3     from locations
4   where country_id in ('CA',
5                         'US')
6   and length(street_address) > 15;
7
8 select first_name,
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

All rows fetched: 2 in 0.011 seconds

	STREET_ADDRESS	CITY
1	2014 Jabberwocky Rd	Southlake
2	2011 Interiors Blvd	South San Francisco

- Muestra la longitud del nombre y el salario anual (por 14) para los empleados cuyo apellido contenga el carácter 'b' después de la 3^a posición.

```

18 select first_name,
19       length(first_name),
20       last_name,
21       salary * 14
22   from employees
23  where instr(
24    last_name,
25    'b'
26  ) > 3;

```

BLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

rows fetched: 4 in 0.014 seconds

	FIRST_NAME	LENGTH(FIRST_NAME)	LAST_NAME	SALARY*14
	Valli	5	Pataballa	67200
	Nancy	5	Greenberg	168112
	Gerald	6	Cambrault	154000
	Nanette	7	Cambrault	105000

- Averiguar los empleados que ganan entre 4000 y 7000 euros y que tienen alguna 'a' en el nombre. (Debemos usar INSTR y da igual que sea mayúscula que minúsculas) y que tengan comisión.

```

18 select *
19   from employees
20  where salary between 4000 and 7000
21    and instr(
22      upper(first_name),
23      'A'
24    ) <> 0
25    and commission_pct is not null;
26
27 select first_name,
28        last_name

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

11 rows fetched: 6 in 0.021 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
1	161	Sarath	Sewall	SSEWALL	011.44.1345.529268
2	165	David	Lee	DLEE	011.44.1346.529268
3	166	Sundar	Ande	SANDE	011.44.1346.629268
4	167	Amit	Banda	ABANDA	011.44.1346.729268
5	173	Sundita	Kumar	SKUMAR	011.44.1343.329268

- Visualizar las iniciales de nombre y apellidos separados por puntos.

The screenshot shows a PostgreSQL query editor with the following code:

```
27 select first_name,
28   ... last_name,
29   ... substr(
30   ...   first_name,
31   ...   1,
32   ...   1
33   ...
34   ... || '. '
35   ... || upper(substr(
36   ...   last_name,
37   ...   1,
38   ...   1
39   ... ))
```

The code uses the `substr` function to extract the first character of each name and concatenates them with a dot. The `upper` function is used to ensure the initials are in uppercase.

Below the code, the interface shows tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, PORTS, and QUERY RESULT. The QUERY RESULT tab is selected, displaying the results of the query:

Fetched 107 rows in 0.053 seconds

	FIRST_NAME	LAST_NAME	INICIALES
1	Ellen	Abel	E.A.
2	Sundar	Ande	S.A.
3	Mozhe	Atkinson	M.A.

- Mostrar empleados donde el nombre o apellido comienza con S..

```
cle > practicas > s06 > s06.sql > ...
3   select first_name,
4       last_name
5   from employees
6   where first_name like 'S%'
7   or last_name like 'S%';
8
9   select first_name,
0       salary,
1       rpad(
2           '*',
3           salary / 1000,
4
OBLEMS     OUTPUT     DEBUG CONSOLE     TERMINAL     PORTS     QUERY RESULT     ...
rows fetched: 20 in 0.024 seconds
```

	FIRST_NAME	LAST_NAME
1	Sundar	Ande
2	Shelli	Baida
3	Sarah	Bell
4	Shelley	Higgins

- Visualizar el nombre del empleado, su salario, y con asteriscos, el número miles de dólares que gana. Se asocia ejemplo. (PISTA: se puede usar RPAD. Ordenado por salario)

```

49  select first_name,
50    salary,
51    rpad(
52      '*' ,
53      salary / 1000,
54      '*'
55    ) as ranking
56  from employees
57  order by salary desc;
58

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

Fetched 107 rows in 0.101 seconds

	FIRST_NAME	SALARY	RANKING
1	Steven	24000	*****
2	Neena	17000	*****
3	Lex	17000	*****
4	John	14000	*****
5	Karen	13500	*****
6	Michael	12000	*****

3.2 Practicas con funciones de Fecha

- Indicar el número de días que los empleados llevan en la empresa

```

8
9  select first_name,
10   hire_date,
11   sysdate - hire_date
12  from employees;
13
14  select sysdate + 15
15   from dual;
16

```

OBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

Fetched 107 rows in 0.036 seconds

	FIRST_NAME	HIRE_DATE	SYSDATE-HIRE_DATE
1	Steven	17/06/03	7877.884675925925925925925925925925925926
2	Neena	21/09/05	7050.884675925925925925925925925925926
3	Lex	13/01/01	8762.884675925925925925925925925925926
4	Alexander	03/01/06	6946.884675925925925925925925925925926
5	Bruce	21/05/07	6443.884675925925925925925925925925926
6	David	25/06/05	7138.884675925925925925925925925925926

- Indicar la fecha que será dentro de 15 días

```

13
14 select sysdate + 15
15   from dual;
16
17 select round(
18       months_between(
19           '25-12-2018',
20           sysdate
21       ),
22
PROBLEMS     OUTPUT     DEBUG CONSOLE    TERMINAL
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75

```

All rows fetched: 1 in 0.013 seconds

	SYSDATE+15
1	24/01/25

- ¿Cuántos MESES faltan para la navidad? La cifra debe salir redondeada, con 1 decimal

```

racine / pratiques / s00 / s00.sql ...
67 select round(
68   months_between(
69     '25-12-2018',
70     sysdate
71   ),
72   1
73 )
74   from dual;
75

```

All rows fetched: 1 in 0.024 seconds

	ROUND(MONTHS_BETWEEN('25-12-2018' ,SYSDATE) ,1)
1	- 72.5

- Indicar la fecha de entrada de un empleado y el último día del mes que entró

The screenshot shows a database interface with two queries and their results.

```
select first_name,
       hire_date,
       last_day(hire_date)
  from employees;
```

```
select first_name,
       hire_date,
       ...
  from employees;
```

Results:

	FIRST_NAME	HIRE_DATE	LAST_DAY(HIRE_DATE)
	Steven	17/06/03	30/06/03
	Neena	21/09/05	30/09/05
	Lex	13/01/01	31/01/01
	Alexander	03/01/06	31/01/06
	Bruce	21/05/07	31/05/07
	David	25/06/05	30/06/05
	Valli	05/02/06	28/02/06

- Utilizando la función ROUND, indicar los empleados que entraron en los últimos 15 días de cada mes

```
81 select first_name,
82      ..... hire_date,
83      ..... round(
84      ..... hire_date,
85      ..... 'MONTH'
86      ..... ) as redondeo
87 from employees
88 where round(
89      ..... hire_date,
90      ..... 'MONTH'
91      ..... hire_date)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

1 rows fetched: 57 in 0.016 seconds

	FIRST_NAME	HIRE_DATE	REDONDEO
1	Steven	17/06/03	01/07/03
2	Neena	21/09/05	01/10/05
3	Bruce	21/05/07	01/06/07
4	David	25/06/05	01/07/05
5	Nancy	17/08/02	01/09/02

4 MODULO 7

4.1 Practicas con la funcion TO CHAR

- Indicar los empleados que entraron en Mayo en la empresa. Debemos buscar por la abreviatura del mes

```

1 select *
2   from employees
3  where to_char(
4    hire_date,
5    'MON'
6  ) = 'MAY';
7
8 select first_name,
9       hire_date
10  from employees
11  where to_char(
12    hire_date

```

rows fetched: 6 in 0.010 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER
1	104	Bruce	Ernst	BERNST	590.423.4568
2	115	Alexander	Khoo	AKHOO	515.127.4562
3	122	Payam	Kaufling	PKAUFLIN	650.123.3234
4	174	Ellen	Abel	EABEL	011.44.1644.429267
5	178	Kimberely	Grant	KGRANT	011.44.1644.429263

Indicar los empleados que entraron en el año 2007, utilizando la función to_char

```
select first_name,  
       hire_date  
  from employees  
 where to_char(  
   hire_date,  
   'YYYY'  
) = 2007;
```

```
select to_char(
```

BLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

rows fetched: 19 in 0.012 seconds

	FIRST_NAME	HIRE_DATE
	Bruce	21/05/07
	Diana	07/02/07
	Luis	07/12/07
	Karen	10/08/07
	Kevin	16/11/07

¿Qué día de la semana en letra era el día que naciste?

```
select to_char(  
    to_date('16-06-1966'),  
    'DAY'  
)  
from dual;  
  
select first_name,  
       hire_date  
  from employees  
 where rtrim(to_char(  
    hire_date,  
    'TO_CHAR(TO_DATE('16-06-1966'), 'DAY')  
    THURSDAY
```

LEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

ws fetched: 1 in 0.011 seconds

TO_CHAR(TO_DATE('16-06-1966'), 'DAY')
THURSDAY

- Averiguar los empleados que entraron en el mes de Junio. Debemos preguntar por el mes en letra. Nota: La función TO_CHAR puede devolver espacios a la derecha)

```
2
3 select first_name,
4       hire_date
5   from employees
6 where rtrim(to_char(
7   hire_date,
8   'MONTH'
9 )) = 'JUNE';
0
1 select salary,
2       to_char(
2      salary
2
2
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ..
```

rows fetched: 11 in 0.011 seconds

	FIRST_NAME	HIRE_DATE
1	Steven	17/06/03
2	David	25/06/05
3	Jason	14/06/04
4	Martha	21/06/07
5	Julia	24/06/06

- Visualizar el salario de los empleados con dos decimales y en dólares y también en la moneda local (el ejemplo es con euros, suponiendo que el cambio esté en 0,79\$)

```
1 select salary,
2      ... to_char(
3      ...     ... salary,
4      ...     '$99,999.99'
5      ... ) as dolares,
6      ... to_char(
7      ...     ... salary * 0.79,
8      ...     'U99,999.99'
9      ... ) as euros
0   from employees;
```

PROMPTS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

checked 107 rows in 0.051 seconds

	SALARY	DOLARES	EUROS
	24000	\$24,000.00	\$18,960.00
	17000	\$17,000.00	\$13,430.00
	17000	\$17,000.00	\$13,430.00
	9000	\$9,000.00	\$7,110.00
	6000	\$6,000.00	\$4,740.00

4.2 Practicas con la funcion TO DATE y TO NUMBER

- Convertir las siguientes cadenas a números

```
2 select to_number('1210.73',
3 ..... '9999.99')
4 from dual;
5
6 select to_number('$127.2',
7 ..... '$999.99')
8 from dual;
9
0 select phone_number,
1 ..... to_number(substr(
2 ..... phone_number,
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

rows fetched: 1 in 0.011 seconds

	TO_NUMBER('1210.73','9999.99')
1	1210.73

```
1 select to_number('$127.2',
2 ..... '$999.99')
3 from dual;
4
5 select phone_number,
6 ..... to_number(substr(
7 ..... phone_number,
```

BLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

rows fetched: 1 in 0.018 seconds

	TO_NUMBER('\$127.2','\$999.99')
	127.2

- Convertir los 3 primeros caracteres del número de teléfono en números y multiplicarlos por 2.

```
49
50     select phone_number,
51           to_number(substr(
52             phone_number,
53             1,
54             3
55           )) * 2
56   from employees;
57
```

fetched 107 rows in 0.044 seconds

	PHONE_NUMBER	TO_NUMBER(SUBSTR(PHONE_NUMBER,1,3))*2
1	515.123.4567	1030
2	515.123.4568	1030
3	515.123.4569	1030
4	590.423.4567	1180
5	590.423.4568	1180

- Convertir las siguientes cadenas en fecha (NOTA: el mes lo debemos poner en el idioma que tengamos en el SqlDeveloper.

```
49
50     select to_date('FACTURA: MARZO0806', '"FACTURA:" MONTHYYDD', 'NLS_DATE_LANGUAGE=SPANISH')
51   from dual;
```

fetched 1 in 0.017 seconds

TO_DATE('FACTURA:MARZO0806', '"FACTURA:"MONTHYYDD', 'NLS_DATE_LANGUAGE=SPANISH')
06/03/08

```
49
50     select to_date('FACTURA: MARZO0806', '"FACTURA:" MONTHYYDD', 'NLS_DATE_LANGUAGE=SPANISH')
51   from dual;
```

fetched 1 in 0.053 seconds

TO_DATE('FACTURA:MARZO0806', '"FACTURA:"MONTHYYDD', 'NLS_DATE_LANGUAGE=SPANISH')
06/03/08

5 MODULO 8

5.1 Practicas con filtrado de Nulos

- De la tabla LOCATIONS visualizar el nombre de la ciudad y el estado-provincia. En el caso de que no tenga que aparezca el texto “No tiene”

```
e>practicas>su8>su8.sql>...
SELECT CITY, NVL(STATE_PROVINCE, 'No tiene') FROM LOCATIONS;
SELECT
SALARY, COMMISSION_PCT, NVL2(COMMISSION_PCT, SALARY+SALARY*COMMISSION_PCT/100, SALARY)
FROM DEPARTMENTS;

SELECT
CITY, STATE_PROVINCE, NVL(CITY, STATE_PROVINCE) FROM
LOCATIONS;
```

LEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR

rows fetched: 23 in 0.011 seconds

CITY	NVL(STATE_PROVINCE, 'NOTIENE')
Roma	No tiene
Venice	No tiene
Tokyo	Tokyo Prefecture
Hiroshima	No tiene
Southlake	Texas
South San Francisco	California

- Visualizar el salario de los empleados incrementado en la comisión (PCT_COMMISION). Si no tiene comisión solo debe salir el salario

```
SELECT
SALARY, COMMISSION_PCT, NVL2(COMMISSION_PCT, SALARY+SALARY*COMMISSION_PCT/100, SALARY) AS TOTAL FROM
DEPARTMENTS;

SELECT
CITY, STATE_PROVINCE, NVL(CITY, STATE_PROVINCE) FROM
LOCATIONS;
```

LEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR COMMENTS

loaded 107 rows in 0.053 seconds

SALARY	COMMISSION_PCT	TOTAL
24000	(null)	24000
17000	(null)	17000
17000	(null)	17000
9000	(null)	9000
6000	(null)	6000

- Seleccionar el nombre del departamento y el manager_id. Si no tiene, debe salir un -1

```

SELECT DEPARTMENT_NAME, NVL(MANAGER_ID, -1) FROM DEPARTMENTS;

SELECT
CITY, STATE_PROVINCE, NULLIF(CITY, STATE_PROVINCE) FROM
LOCATIONS;

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR COMMENTS ≡ ^

rows fetched: 27 in 0.015 seconds

DEPARTMENT_NAME	NVL(MANAGER_ID, -1)
Administration	200
Marketing	201
Purchasing	114
Human Resources	203
Shipping	121
IT	103

- De la tabla LOCATIONS, devolver NULL si la ciudad y la provincia son iguales. Si no son iguales devolver la CITY.

```

SELECT
CITY, STATE_PROVINCE, NULLIF(CITY, STATE_PROVINCE) FROM
LOCATIONS;

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR COMMENTS

rows fetched: 23 in 0.020 seconds

CITY	STATE_PROVINCE	NULLIF(CITY, STATE_PROVINCE)
Roma	(null)	Roma
Venice	(null)	Venice
Tokyo	Tokyo Prefecture	Tokyo
Hiroshima	(null)	Hiroshima
Southlake	Texas	Southlake

6 MODULO 9

6.1 Practicas con expresiones condicionales

- Visualizar los siguientes datos con CASE.
 - o Si el departamento es 50 ponemos Transporte
 - o Si el departamento es 90 ponemos Dirección
 - o Cualquier otro número ponemos “Otro departamento”



```
le > practicas > s09 > s09.sql > ...
select department_id,
       first_name,
       case department_id
           when 50 then
               'Transporte'
           when 60 then
               'Dirección'
           else
               'Otro departamento'
       end
```

BLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

ched 107 rows in 0.050 seconds

	DEPARTMENT_ID	FIRST_NAME	CASEDEPARTMENT_IDWHEN50THEN 'TRANSPORTE'
	90	Steven	Otro departamento
	90	Neena	Otro departamento
	90	Lex	Otro departamento
	60	Alexander	Dirección
	60	Bruce	Dirección
	60	David	Dirección

- Mostrar de la tabla LOCATIONS, la ciudad y el país. Ponemos los siguientes datos dependiendo de COUNTRY_ID.
 - o Si es US y CA ponemos América del Norte
 - o Si es CH, UK, DE,IT ponemos Europa
 - o Si es BR ponemos América del Sur
 - o Si no es ninguno ponemos ‘Otra zona’

```
select country_id,  
       city,  
       case  
           when country_id in ( 'US',  
                                 'CA' ) then  
               'AMERICA DEL NORTE'  
           when country_id in ( 'CH',  
                                 'UK',  
                                 'DE',  
                                 'ES' ) then  
               'EUROPA'  
           else  
               'OTRA ZONA'  
       end  
       as CASEWHENCOUNTRY_IDIN('US', 'CA')TH  
from cities
```

WS fetched: 23 in 0.011 seconds

COUNTRY_ID	CITY	CASEWHENCOUNTRY_IDIN('US', 'CA')TH
IT	Roma	EUROPA
IT	Venice	EUROPA
JP	Tokyo	OTRA ZONA
JP	Hiroshima	OTRA ZONA
US	Southlake	AMERICA DEL NORTE
US	South San Francisco	AMERICA DEL NORTE

- Realizar el primer ejercicio con DECODE en vez de con CASE

```
select department_id,  
       first_name,  
       decode(  
           department_id,  
           50, 'Transporte',  
           60, 'Dirección',  
           'Otro departamento'  
)
```

BLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

QUERY RESULT

...

ched 107 rows in 0.028 seconds

	DEPARTMENT_ID	FIRST_NAME	DECODE(DEPARTMENT_ID, 50, 'TRA
	90	Steven	Otro departamento
	90	Neena	Otro departamento
	90	Lex	Otro departamento
	60	Alexander	Dirección
	60	Bruce	Dirección
	60	David	Dirección

7 MODULO 10

7.1 Funciones con grupos

- Indicar el número de empleados del departamento 50

```
> /prácticas /st0 / st0.sql > ...  
select count(*)  
from employees  
where department_id = 50;  
  
select count(*)  
from employees  
where to_char(  
    hire_date,  
    'yyyy'  
) = '2007';
```

ROWS fetched: 1 in 0.022 seconds

COUNT(*)
45

- Indicar el número de empleados que entraron en el año 2007 a trabajar

```
select count(*)  
from employees  
where to_char(  
    hire_date,  
    'yyyy'  
) = '2007';  
  
select max(salary),  
       min(salary),
```

ROWS fetched: 1 in 0.017 seconds

COUNT(*)
19

- Indicar la diferencia entre el sueldo más alto y al mínimo

```
select max(salary),  
.....|... min(salary),  
.....|... max(salary) - min(salary)  
..from employees;  
  
select sum(salary)  
  from employees  
 where department_id = 100;  
  
select department_id,
```

1s fetched: 1 in 0.011 seconds

MAX(SALARY)	MIN(SALARY)	MAX(SALARY)-MIN(SALARY)
24000	2100	21900

- Visualizar la suma del salario del departamento 100

```
select sum(salary)  
..from employees  
where department_id = 100;  
  
select department_id,  
       round(  
           avg(salary),  
           2  
       )
```

1s fetched: 1 in 0.012 seconds

SUM(SALARY)
51608

- Mostrar el salario medio por departamento, con dos decimales

```
select department_id,
       round(
           avg(salary),
           2
       )
  from employees
 group by department_id;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

rows fetched: 12 in 0.059 seconds

DEPARTMENT_ID	ROUND(AVG(SALARY), 2)
100	8601.33
30	4150
(null)	7000
90	19333.33
20	9500

- Mostrar el country_id y el número de ciudades que hay en ese país.

```
select country_id,
       count(*)
  from locations
 group by country_id;

select department_id,
       round(
           avg(salary),
           2
       )
  from employees
 group by department_id;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

rows fetched: 14 in 0.138 seconds

COUNTRY_ID	COUNT(*)
US	4
SG	1
CA	2
CH	2
IT	2

- Mostrar el promedio de salario de los empleados por departamento que tengan comisión

```

select department_id,
       round(
           avg(salary),
           2
       )
  from employees
 where commission_pct is not null
 group by department_id;

```

BLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

rows fetched: 2 in 0.122 seconds

DEPARTMENT_ID	ROUND(AVG(SALARY), 2)
(null)	7000
80	8955.88

- Mostrar los años en que ingresaron más de 10 empleados

```

select to_char(
  ... hire_date,
  ... 'YYYY'
),
       count(*)
  from employees
 group by to_char(
  ... hire_date,
  ... 'YYYY'
)

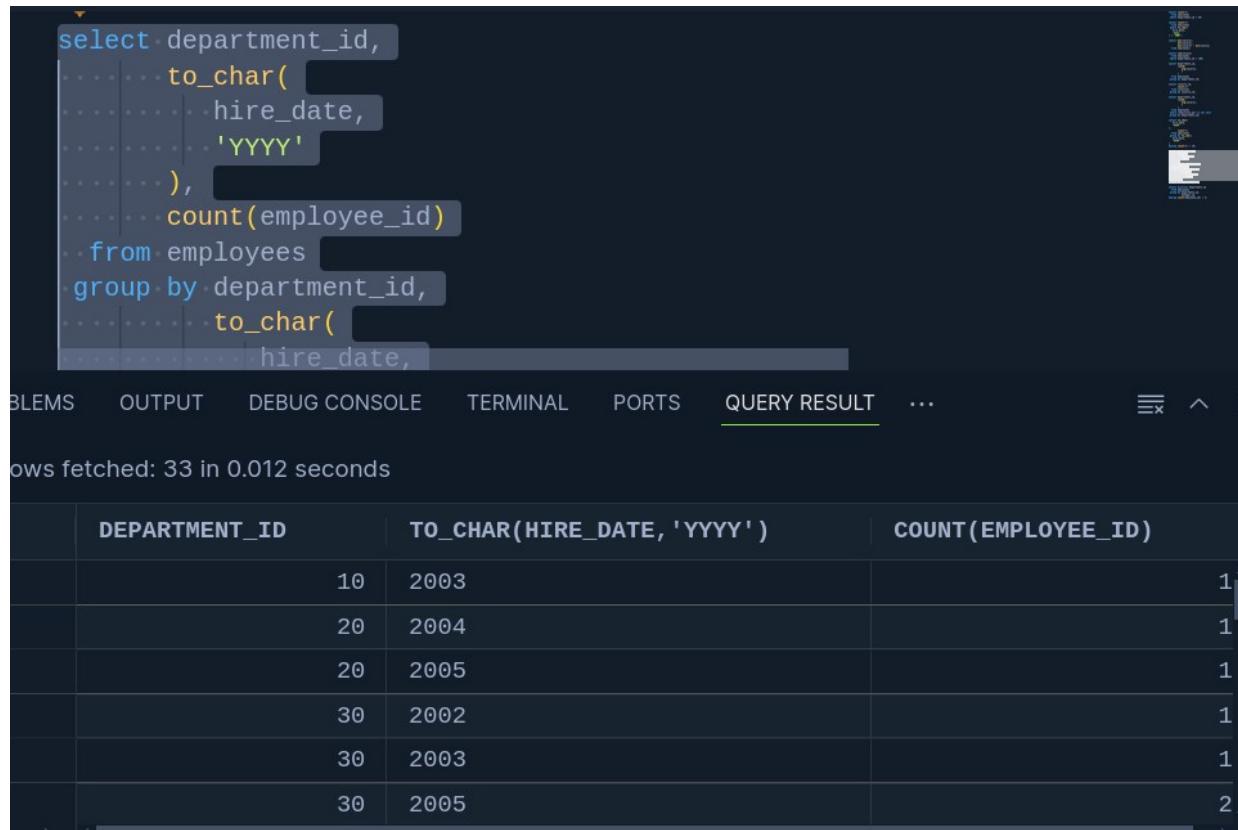
```

BLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT ...

rows fetched: 4 in 0.114 seconds

TO_CHAR(HIRE_DATE, 'YYYY')	COUNT(*)
2005	29
2008	11
2007	19
2006	24

- Mostrar por departamento y año el número de empleados que ingresaron



```

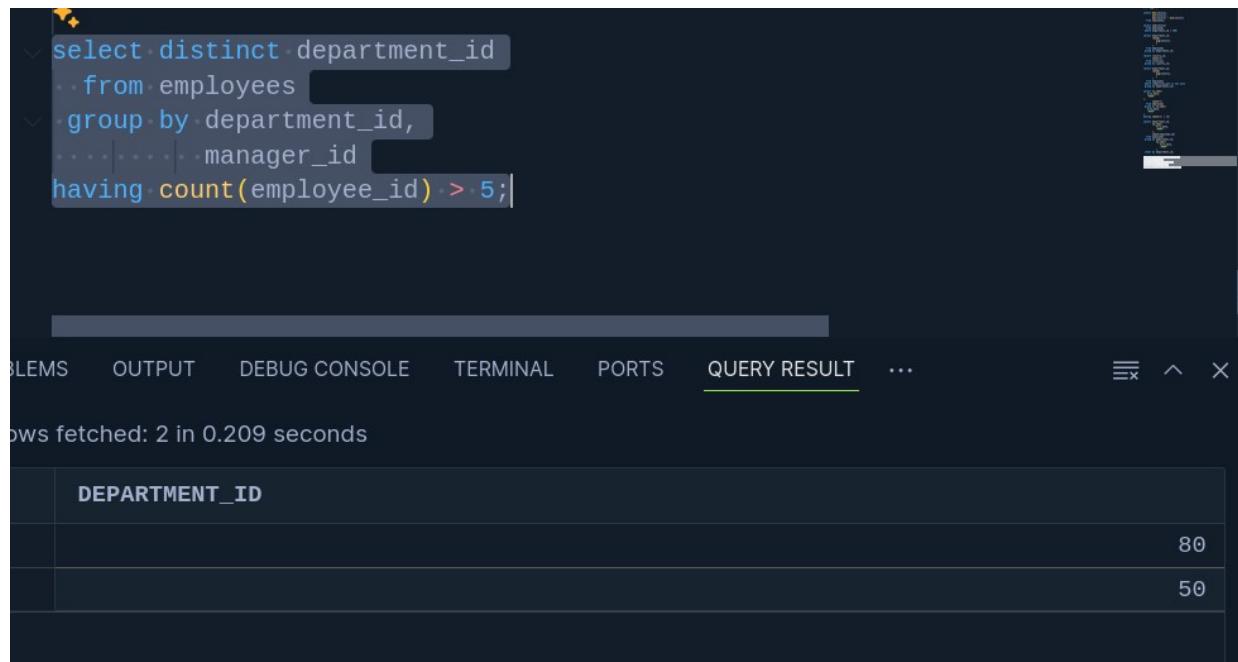
select department_id,
       to_char(
           hire_date,
           'YYYY'
       ),
       count(employee_id)
  from employees
 group by department_id,
          to_char(
              hire_date,

```

ROWS Fetched: 33 in 0.012 seconds

DEPARTMENT_ID	TO_CHAR(HIRE_DATE, 'YYYY')	COUNT(EMPLOYEE_ID)
10	2003	1
20	2004	1
20	2005	1
30	2002	1
30	2003	1
30	2005	2

- Mostrar los department_id de los departamentos que tienen managers que tienen a cargo más de 5 empleados



```

select distinct department_id
  from employees
 group by department_id,
          manager_id
 having count(employee_id) > 5;

```

ROWS Fetched: 2 in 0.209 seconds

DEPARTMENT_ID
80
50

8 MODULO 11

8.1 Practicas con Joins-Natural

- Visualizar el nombre del país y el nombre de la región. (tablas COUNTRIES y REGIONS). Usar un natural join

```
oracle > practicas > s11 > s11.sql > ...
1  select countries.country_name,
2      region_name
3  from regions
4  natural join countries;
5
6  select countries.country_name,
7         region_name,
8         city
9  from regions
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASKS

All rows fetched: 25 in 0.122 seconds

	COUNTRY_NAME	REGION_NAME
1	Netherlands	Europe
2	France	Europe
3	United Kingdom	Europe
4	Denmark	Europe
5	Belgium	Europe
6	Switzerland	Europe

- Usando el ejemplo anterior visualizar también el nombre de la ciudad añadiendo una nueva tabla (LOCATIONS)

```

3 select countries.country_name,
4     ....region_name,
5     ....city
6     from regions
7 natural join countries
8 natural join locations;

9 select department_name.
10
11 PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   QUERY RESULT   SQL HISTORY   TASK MONITOR   COMMENTS
12
13 rows fetched: 23 in 0.155 seconds

```

COUNTRY_NAME	REGION_NAME	CITY
United Kingdom	Europe	Stretford
United Kingdom	Europe	Oxford
United Kingdom	Europe	London
Netherlands	Europe	Utrecht
Italy	Europe	Venice
Italy	Europe	Roma

- Indicar el nombre del departamento y la media de sus salarios

```

3 select department_name,
4     ....round(
5     ........avg(salary),
6     ....2
7     ....)
8     from employees
9     join departments
10    using (department_id)
11    group by department_name;
12
13 PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   QUERY RESULT   SQL HISTORY   TASK MONITOR   COMMENTS
14
15 rows fetched: 11 in 0.128 seconds

```

DEPARTMENT_NAME	ROUND(AVG(SALARY),2)
Administration	4400
Accounting	10154
Purchasing	4150
Human Resources	6500
IT	5760
Public Relations	10000
Executive	19333.33

- Mostrar el nombre del departamento, el del manager a cargo y la ciudad a la que pertenece. Debemos usar la cláusula ON y/o la cláusula USING para realizar la operación

```

oracle > practicas > s11 > s11.sql > ...
23  select department_name,
24    .....first_name,
25    .....city
26  from departments dept
27  join employees emple
28  on ( dept.manager_id = emple.employee_id )
29  join locations loc
30  using ( location_id );
31

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR COMMENTS

All rows fetched: 11 in 0.112 seconds

	DEPARTMENT_NAME	FIRST_NAME	CITY
1	Public Relations	Hermann	Munich
2	Shipping	Adam	South San Francisco
3	Finance	Nancy	Seattle
4	Marketing	Michael	Toronto
5	Accounting	Shelley	Seattle
6	IT	Alexander	Southlake
7	Executive	Steven	Seattle

- Mostrar job_title, el department_name, el last_name de empleado y hire_date de todos los empleados que entraron entre el 2000 y el 2004.
Usar cláusulas using

```

2  select last_name,
3    .....hire_date,
4    .....department_name,
5    .....job_title
6  from employees
7  join departments
8  using ( department_id )
9  join jobs

```

OBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR COMMENTS

rows fetched: 23 in 0.112 seconds

	LAST_NAME	HIRE_DATE	DEPARTMENT_NAME	JOB_TITLE
1	King	17/06/03	Executive	President
2	Greenberg	17/08/02	Finance	Finance Manager
3	Faviet	16/08/02	Finance	Accountant
4	Raphaely	07/12/02	Purchasing	Purchasing Manager
5	Khoo	18/05/03	Purchasing	Purchasing Clerk
6	Weiss	18/07/04	Shipping	Stock Manager
7	Kaufling	01/05/03	Shipping	Stock Manager

- Mostrar el job_title y la media de los salarios de cada uno, siempre que la media supere los 7000

```

46 select.job_title,
47 .....avg(salary)
48 ..from jobs
49 natural join employees
50 .group by job_title
51 having avg(salary) > 7000;
52
53 select region_name,
54 | | count(*) as "NUM DEPAR"

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR COMMENTS  

11 rows fetched: 12 in 0.078 seconds

	JOB_TITLE	AVG(SALARY)
1	Accounting Manager	12008
2	Public Relations Representative	10000
3	Sales Representative	8350
4	Administration Vice President	17000
5	Stock Manager	7280
6	President	24000
7	Finance Manager	12008

- Mostrar el nombre de la región y el número de departamentos en cada una de las regiones

```

select.region_name,
.....count(*) as "NUM DEPAR"
..from regions
natural join countries
natural join locations
natural join departments
| group by region_name;

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR COMMENTS  

2 rows fetched: 2 in 0.211 seconds

	REGION_NAME	NUM DEPAR
	Europe	3
	Americas	24

- Mostrar el nombre del empleado, el departamento y el país donde trabaja (debemos usar la cláusula using)

```

select first_name,
       department_name,
       country_name
  from employees
  join departments
using (department_id)
  join locations
using (location_id)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR COMMENTS

selected 106 rows in 0.188 seconds

	FIRST_NAME	DEPARTMENT_NAME	COUNTRY_NAME
	Ellen	Sales	United Kingdom
	Sundar	Sales	United Kingdom
	Mozhe	Shipping	United States of America
	David	IT	United States of America
	Hermann	Public Relations	Germany
	Shelli	Purchasing	United States of America
	Amit	Sales	United Kingdom

8.2 Prácticas con otros joins

- Indicar el nombre del empleado y el de su jefe (SELF_JOIN de la tabla EMPLOYEES)

```

select emple.first_name as empleado,
       jefes.first_name as jefe
  from employees emple
       join employees jefes
    on emple.manager_id = jefes.employee_id;

select department_name,
       sum(salary) as num_emple

```

PROMPTS OUTPUT DEBUG CONSOLE TERMINAL PORTS **QUERY RESULT** SQL HISTORY TASKS

Rows fetched 106 rows in 0.271 seconds

EMPLEADO	JEFE
William	Gerald
Lisa	Gerald
Sundita	Gerald
Tayler	Gerald
Harrison	Gerald
Elizabeth	Gerald
Alexander	Lex

- Indica el DEPARTMENT_NAME y la suma de salarios de ese departamento ordenados ascendente y que aparezcan también los DEPARTMENT_NAME que no tengan empleados.

```

select department_name,
       sum(salary) as num_emple
  from employees
 right outer join departments
 using ( department_id )
 group by department_name
 order by sum(salary);

```

PROMPTS OUTPUT DEBUG CONSOLE TERMINAL PORTS **QUERY RESULT** SQL HISTORY TASK MONITOR COMMENTS

Rows fetched: 27 in 0.063 seconds

DEPARTMENT_NAME	NUM_EMPL
Administration	4400
Human Resources	6500
Public Relations	10000
Marketing	19000
Accounting	20308
Purchasing	24900
IT	28800

- Visualizar la ciudad y el nombre del departamento, incluidas aquellas ciudades que no tengan departamentos

```
5
6 select city,
7      department_name
8   from locations
9  left join departments
0 using ( location_id );|
```

rows fetched: 43 in 0.139 seconds

	CITY	DEPARTMENT_NAME
1	Seattle	Administration
2	Toronto	Marketing
3	Seattle	Purchasing
4	London	Human Resources
5	South San Francisco	Shipping
6	Southlake	IT
7	Munich	Public Relations

9 MODULO 12

9.1 Practicas con subconsultas

- Mostrar los compañeros que trabajan en el mismo departamento que John Chen

```
1 select first_name,  
2     last_name  
3 from employees  
4 where department_id = (  
5 select department_id  
6 from employees  
7 where first_name = 'John'  
8 and last_name = 'Chen'  
9 );
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY T

11 rows fetched: 6 in 0.035 seconds

	FIRST_NAME	LAST_NAME
1	Nancy	Greenberg
2	Daniel	Faviet
3	John	Chen
4	Ismael	Sciarra
5	Jose Manuel	Urman
6	Luis	Ranger
7	Ma	Solomon
8	Pat	Solomon
9	Susan	Solomon
10	Valli	Solomon

- ¿Qué departamentos tienen su sede en Toronto?

```
1 select department_name  
2 from departments  
3 where location_id = (  
4 select location_id  
5 from locations  
6 where city = 'Toronto'  
7 );
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY T

1 rows fetched: 1 in 0.032 seconds

DEPARTMENT_NAME
Marketing

- Visualizar los empleados que tengan más de 5 empleados a su cargo.

The screenshot shows a SQL query in the SQL worksheet of Oracle SQL Developer. The code is as follows:

```
19 select first_name  
20   from employees  
21  where employee_id in (  
22    select manager_id  
23     from employees  
24    group by manager_id  
25   having count(*) > 5  
26 );  
27
```

Below the code, the output window displays:

```
11 rows fetched: 11 in 0.071 seconds
```

	FIRST_NAME
1	Steven
2	Adam
3	Payam
4	Alberto
5	John
6	Eleni

- ¿En qué ciudad trabajar Guy Himuro?

```
oracle > practicas > STZ > STZ.sql > ...
28  select city
29    from locations
30   where location_id = (
31     select location_id
32       from departments
33      where department_id = (
34        select department_id
35          from employees
36         where first_name = 'Guy'
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

All rows fetched: 1 in 0.013 seconds

	CITY
1	Seattle

- ¿Qué empleados tienen el salario mínimo?

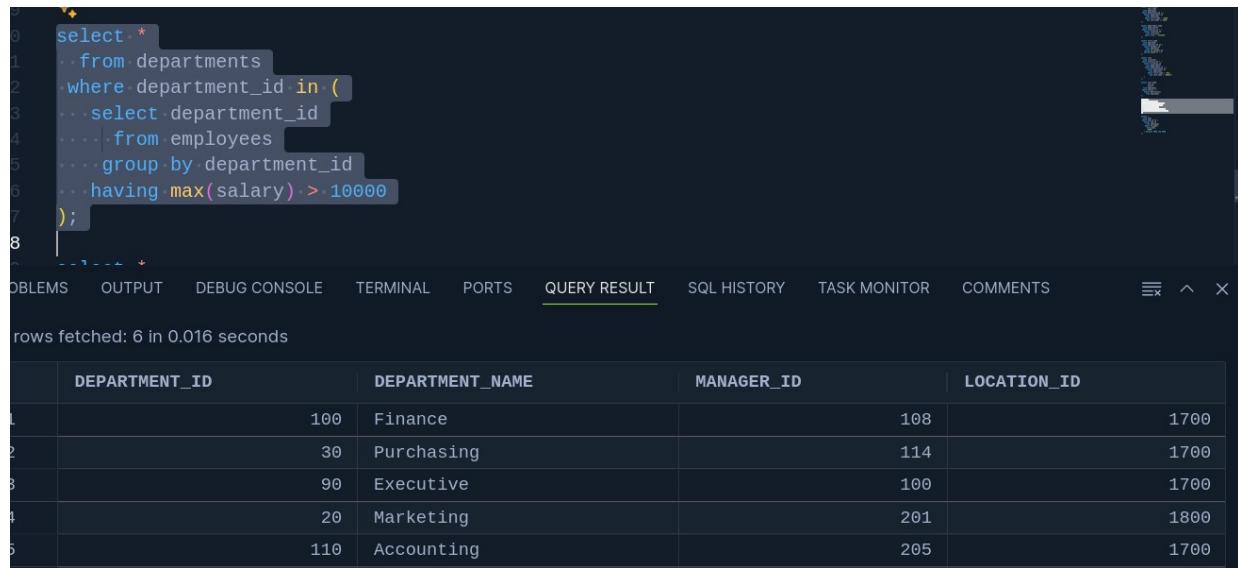
```
oracle > practicas > STZ > STZ.sql > ...
1  select last_name,
2    job_id,
3    salary
4  from employees
5  where salary = (
6    select min(salary)
7    from employees
8  );
9
10 select *
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR COMMENTS

All rows fetched: 1 in 0.012 seconds

	LAST_NAME	JOB_ID	SALARY
1	Olson	ST_CLERK	2100

- Mostrar los detalles de los departamentos en los cuales el salario máximo sea mayor a 10000.



```

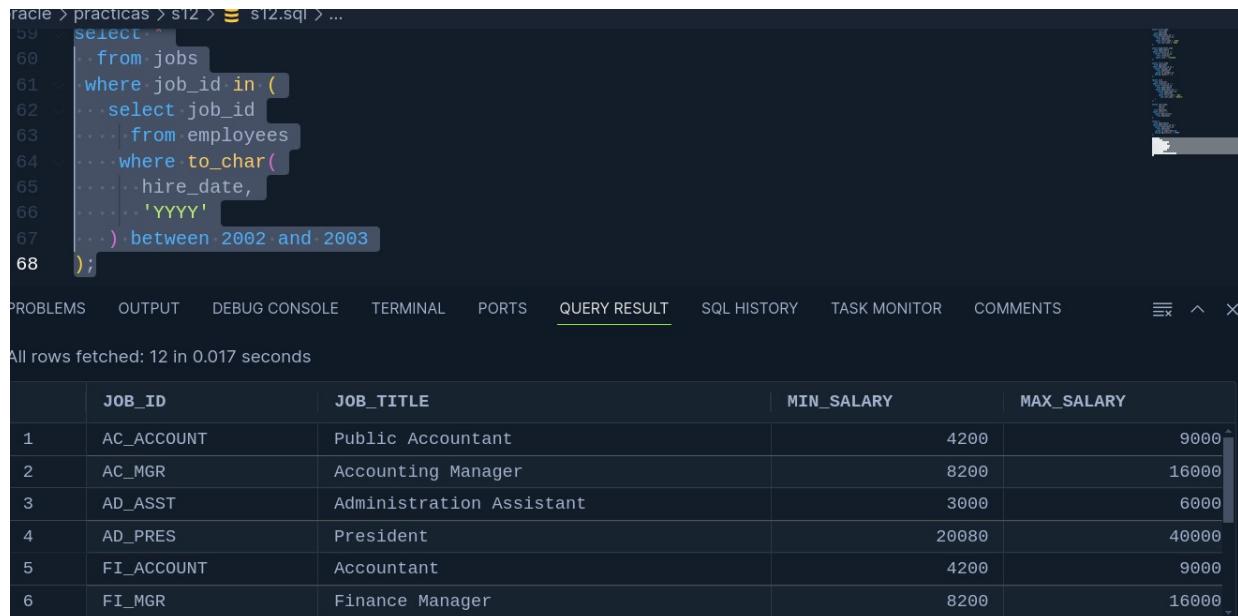
5
6 select *
7   from departments
8  where department_id in (
9    select department_id
10   from employees
11   group by department_id
12   having max(salary) > 10000
13 );
14
15

```

rows fetched: 6 in 0.016 seconds

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	100	Finance	108	1700
2	30	Purchasing	114	1700
3	90	Executive	100	1700
4	20	Marketing	201	1800
5	110	Accounting	205	1700

- Indicar los tipos de trabajo de los empleados que entraron en la empresa entre 2002 y 2003



```

59
60 select *
61   from jobs
62  where job_id in (
63    select job_id
64      from employees
65     where to_char(
66       hire_date,
67       'YYYY'
68     ) between 2002 and 2003
69 );
70
71

```

All rows fetched: 12 in 0.017 seconds

	JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
1	AC_ACCOUNT	Public Accountant	4200	9000
2	AC_MGR	Accounting Manager	8200	16000
3	AD_ASST	Administration Assistant	3000	6000
4	AD_PRES	President	20080	40000
5	FI_ACCOUNT	Accountant	4200	9000
6	FI_MGR	Finance Manager	8200	16000

9.2 Practicas con otras subconsultas

- Seleccionar el nombre, salario y departamento de los empleados que ganen mas que cualquiera de los salarios máximos de los departamentos 50, 60 y 70. Usar el operador ANY

```
select first_name,
       salary,
       department_id
  from employees
 where salary > any (
   select max(salary)
     from employees
    group by department_id
   having department_id in ('50'))
```

LEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR COMMENTS

ws fetched: 31 in 0.084 seconds

FIRST_NAME	SALARY	DEPARTMENT_ID
Steven	24000	90
Neena	17000	90
Lex	17000	90
John	14000	80
Karen	13500	80
Michael	13000	20

- Indicar el nombre de los departamentos cuyo salario medio sea superior a 9000. Usar el operador IN

```
select department_name
  from departments
 where department_id in (
   select department_id
     from employees
    group by department_id
   having avg(salary) > 9000
);
```

```
select first_name,
```

LEMS OUTPUT DEBUG CONSOLE TERMINAL

ws fetched: 4 in 0.031 seconds

DEPARTMENT_NAME
Executive
Marketing
Public Relations
Accounting

- Indicar el nombre del empleado, el nombre del departamento, el salario de los empleados que tengan el salario máximo de su departamento. Ordenado por salario descendente. Usar el operador IN

```

2 select first_name,
3       department_name,
4       salary
5   from employees
6  join departments
7     using ( department_id )
8  where ( salary ) in (
9    select department_id,
10       max(salary)
11      )

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR COMMENTS

rows fetched: 11 in 0.025 seconds

	FIRST_NAME	DEPARTMENT_NAME	SALARY
1	Steven	Executive	24000
2	John	Sales	14000
3	Michael	Marketing	13000
4	Nancy	Finance	12008
5	Shelley	Accounting	12008
6	Den	Purchasing	11000

- Realizar la misma consulta anterior pero usando una subconsulta sincronizada

```

07 select first_name,
08       department_name,
09       salary
10  from employees emp
11  join departments dept
12  on ( emp.department_id = dept.department_id )
13  where salary = (
14    select max(salary)
15    from employees
16    where department_id = emp.department_id

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR COMMENTS

rows fetched: 11 in 0.014 seconds

	FIRST_NAME	DEPARTMENT_NAME	SALARY
1	Steven	Executive	24000
2	John	Sales	14000
3	Michael	Marketing	13000
4	Nancy	Finance	12008
5	Shelley	Accounting	12008
6	Den	Purchasing	11000

- Indicar los datos de los empleados que ganen más que todos los empleados del departamento 100. Usar el operador ALL

```
select *
  from employees
 where salary > all (
   select salary
     from employees
    where department_id = 100
);

select department_id,
       ...

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR COMMENTS

rows fetched: 6 in 0.049 seconds

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID
201	Michael	Hartstein	MHARTSTE	515.123.5555	17/02/04	MK_MAN
146	Karen	Partners	KPARTNER	011.44.1344.467268	05/01/05	SA_MAN
145	John	Russell	JRUSSEL	011.44.1344.429268	01/10/04	SA_MAN
102	Lex	De Haan	LDEHAAN	515.123.4569	13/01/01	AD_VP
101	Neena	Kochhar	NKOCHHAR	515.123.4568	21/09/05	AD_VP
100	Steven	King	SKING	515.123.4567	17/06/03	AD_PRES

- Mostrar los empleados que tienen el mayor salario de su departamento. Usar subconsultas sincronizadas.

```
ne /practicas /s12 / s12.sql / ...
select department_id,
       first_name,
       salary
  from employees emp
 where salary = (
   select max(salary)
     from employees
    where department_id = emp.department_id
);


```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SQL HISTORY TASK MONITOR COMMENTS

rows fetched: 11 in 0.023 seconds

DEPARTMENT_ID	FIRST_NAME	SALARY
100	Nancy	12008
30	Den	11000
90	Steven	24000
20	Michael	13000
70	Hermann	10000
110	Shelley	12008

- Visualizar las ciudades en las que haya algún departamento. Debemos usar consultas sincronizadas y el operador EXISTS

```
select city
  from locations.localidades
 where exists (
  select *
    from departments
   where location_id = localidades.location_id
);

select region_name
  from regions.regiones
```

BLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT

rows fetched: 7 in 0.013 seconds

CITY
Southlake
South San Francisco
Seattle
Toronto
London

- Visualizar el nombre de las regiones donde no hay departamentos. Usar subconsultas sincronizadas y el operador NOT EXISTS

```
select region_name
  from regions regiones
 where not exists (
   select *
     from countries
   natural join locations
   natural join departments
    where region_id = regiones.region_id
);
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Q

rows fetched: 2 in 0.089 seconds

REGION_NAME
Asia
Middle East and Africa

10 MODULO 13

10.1 Practicas con operadores de conjunto

- Copiar y ejecutar en el SqlDeveloper las siguientes líneas. Debe crear 2 tablas llamadas coches1 y coches2 y cargarla con datos. En un capítulo posterior veremos como crear tablas y como insertar datos en las mismas. Por ahora nos limitamos a crearlas

```
oracle > practicas > s13 > s13.sql > ...
1  < create table coches1 (
2    codigo number,
3    nombre varchar2(100)
4  );
5
6  < create table coches2 (
7    code number,
8    name varchar2(100)
9  );
10 --INSERTAR DATOS EN TABLAS
11 < insert into coches1 values ( 1,
12   ..... 'BMW 3' );
13 < insert into coches1 values ( 2,
14   ..... 'AUDI A5' );
15 < insert into coches1 values ( 3,
16   ..... 'CITROEN C5' );
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SCRIP

Table COCHES1 creado.

Table COCHES2 creado.

- Probar los 4 operadores de conjunto: UNION, UNION ALL, INTSERSCT, MINUS

```
select *  
from coches1  
union  
select *  
from coches2;  
  
select *  
from coches1  
union all  
select *  
from coches2;
```

BLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SCRIPT OUTPUT SQL HISTO

rows fetched: 6 in 0.025 seconds

CODIGO	NOMBRE
1	BMW 3
2	AUDI A5
3	CITROEN C5
4	RENAULT CLIO
6	MERCEDES C
7	FORD MUSTANG

```
1 select *
2   from coches1
3 union all
4 select *
5   from coches2;
6
7 select *
8   from coches1
9 intersect
10 select *
11
12
```

rows fetched: 8 in 0.015 seconds

CODIGO	NOMBRE
1	BMW 3
2	AUDI A5
3	CITROEN C5
4	RENAULT CLIO
1	BMW 3
6	MERCEDES C

```
1 select *
2   from coches1
3 intersect
4 select *
5   from coches2;
6
7 select *
8   from coches1
9 minus
10 select *
11
```

rows fetched: 2 in 0.012 seconds

CODIGO	NOMBRE
1	BMW 3
3	CITROEN C5

```
FROM COCHES2,  
select *  
from coches1  
minus  
select *  
from coches2;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SCRIPT OUTPUT

rows fetched: 2 in 0.012 seconds

	CODIGO	NOMBRE
	2	AUDI A5
	4	RENAULT CLIO

11 MODULO 14

11.1 Practicas DML

Insertar los siguientes datos en la tabla, indicando todas las columnas

```
1  create table productos (
2      codigo number not null,
3      nombre varchar2(100) not null,
4      precio number not null,
5      unidades number,
6      fecha_alta date
7  );
8
9  insert into productos (
10     codigo,
11     nombre,
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Table PRODUCTOS created.

- Insertar en la tabla PRODUCTOS2 las filas de la tabla PRODUCTOS que tengan más de 8 unidades. Comprobar las filas

```
  insert into productos (
    ...codigo,
    ...nombre,
    ...precio,
    ...unidades,
    ...fecha_alta
  ) values ( 1,
    ...          'tornillos',
    ...          100,
    ...          10)
```

BLEMS OUTPUT DEBUG CONSOLE TERM

Table PRODUCTOS created.

1 fila insertadas.

- Modificar el campo NOMBRE de la tabla PRODUCTOS y poner en mayúsculas el nombre de aquellas filas que valgan más de 50.
 - o Comprobar el resultado

The screenshot shows a MySQL command-line interface with the following code:

```
20
21 ✓ insert into productos values ( 2,
22   ..... | ..... | ..... | ..... | ..... | ..... |
23   ..... | ..... | ..... | ..... | ..... | ..... |
24   ..... | ..... | ..... | ..... | ..... | ..... |
25   ..... | ..... | ..... | ..... | ..... | ..... |
26
27 ✓ insert into productos (
28   |   codigo,
29   |   nombre
```

Below the code, the terminal output is displayed:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS SCRIPT C

Table PRODUCTOS created.

1 fila insertadas.

1 fila insertadas.

```
26  insert into productos (
27    ...codigo,
28    ...nombre,
29    ...precio
30  ) values ( 3,
31              ... 'Martillos',
32              ... 90 );
33
34
35  insert into productos (
36    codigo,
37
38 1 fila insertadas
34
35  insert into productos (
36    ...codigo,
37    ...nombre,
38    ...unidades,
39    ...precio
40  ) values ( 4,
41              ... 'Arandela',
42              ... 10,
43              ... 80 );
44
45  insert into productos (
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

https://docs.oracle.com/error-help/db/oracle-error-00001

More Details :

https://docs.oracle.com/error-help/db/oracle-error-00001

1 fila insertadas.

```
54
55  create table productos2 (
56    code number,
57    name varchar2(100)
58 );
59
60  insert into productos2
61      select codigo,
62            nombre
63      from productos
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL
```

```
1 fila insertadas.
59
60  insert into productos2
61      select codigo,
62            nombre
63      from productos
64      where unidades > 8;
65
66  select *
67      from productos2;
68
69  update productos
70      set
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL
```

1 fila insertadas.

Table PRODUCTOS2 creado.

3 filas insertadas.

```
6 select *
7   from productos2;
8
9 update productos
10    set
11      nombre = upper(nombre)
12     where precio > 50;
13
14 update productos
15    set
16      precio = precio + 5
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULT SCRIPT OUTPUT

rows fetched: 3 in 0.031 seconds

	CODE	NAME
1		1 tornillos
2		4 Arandela
3		4 Arandela

- Modificar el precio de la tabla productos de aquellas filas cuyo nombre comienza por ‘T’. Debemos incrementarlo en 5. Comprobar el resultado

```
update productos
    set nombre = upper(nombre)
   where precio > 50;

update productos
    set
      precio = precio + 5

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL
```

Role PRODUCTOS2 creado.

1 filas insertadas.

1 filas actualizadas.

```
1 update productos
2   set ...
3     precio = precio + 5
4   where upper(nombre) like 'T%';
5
6 delete productos
7   where unidades < 10
8     or unidades is null;
9
10 truncate table productos2;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

filas insertadas.

filas actualizadas.

filas actualizadas.

- Borrar las filas de la tabla productos que tengan menos de 10 unidades o un valor nulo. Comprobar el resultado

```
78
79 delete productos
80 where unidades < 10
81 or unidades is null;
82
83 truncate table productos2;
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

5 filas actualizadas.

2 filas actualizadas.

3 filas eliminado

- Truncar la tabla PRODUCTOS2. Comprobar el resultado

```
82
83 truncate table productos2;|
```

PROBLEMS OUTPUT DEBUG CONSOLE TEP

2 filas actualizadas.

3 filas eliminado

Table PRODUCTOS2 truncado.

12 MODULO 15

12.1 Practicas con Transacciones

- Regresemos al primer SqlDeveloper
 - o Hacer un DELETE de la fila
 - o Comprobar que se ha borrado
 - o Realizar un ROLLBACK
 - o Comprobar que se ha recuperado.

```
oracle> practices> 310> 310.sql> ...
1  insert into productos2 (
2    code,
3    name
4  ) values ( 1,
5              'CLAVOS' );
6
7  insert into productos2 (
8    code,
9    name
10 ) values ( 2,
11               'ARANDELAS' );
12
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
7  insert into productos2 (
8    code,
9    name
10 ) values ( 2,
11               'ARANDELAS' );
12
13 insert into productos2 (
14   code,
15   name
16 ) values ( 3,
17               'ESCARPIAS' );
18
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

1 fila insertadas.

1 fila insertadas.

```
12
13 ✓ insert into productos2 (
14     ... code,
15     ... name
16 ✓ ) values ( 3,
17     ..... | ..... 'ESCARPIAS' .. );
18
19 savepoint a;
20
21 ✓ update productos2
22     set
23         name = 'TORNILLOS'
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TE

1 fila insertadas.

1 fila insertadas.

```
18
19     savepoint a;
20
21     update productos2
22         set
23             name = 'TORNILLOS'
24         where code = 3;
25
26     rollback to a;
```

PROBLEMS OUTPUT DEBUG CONSO

1 fila insertadas.

1 fila insertadas.

Savepoint creado.

```
20
21 ✓ update productos2
22   | set
23   |   name = 'TORNILLOS'
24   | where code = 3;
25
26 rollback to a;
```

PROBLEMS OUTPUT DEBUG CONSO

1 fila insertadas.

Savepoint creado.

1 fila actualizadas.

- Vamos a probar un SAVEPOINT
- o Realizamos estas dos operaciones

```
25
26 rollback to a;

PROBLEMS    OUTPUT    DEBUG C

Savepoint creado.

1 fila actualizadas.

Rollback terminado.
```