UPA

UNIVERSIDAD POLITÉCNICA
DE AGUASCALIENTES

ISC: Ingeniería en Sistemas Computacionales.

Administración de Base de Datos

20-1. Asegurando la calidad de los Resultados de las Consultas

Miembros:

LIZBETH GUADALUPE MORENO JIMENEZUP200738

ROGELIO TADEO GÓMEZ MUÑOZ UP200920

FERNANDO YAEL DE LA TORRE RANGEL UP200994

VIVIAN ANDREA GALLEGOS RODARTE UP200045

ROCIO PAULINA GONZALEZ MORALES UP200604

CESAR ANTONIO DIAZ MORALES UP200953

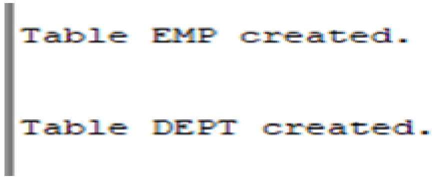
ISC06B.

Profesor: Juan Carlos Herrera Hernández

Aguascalientes, Aguascalientes. 12/11/2022

1.

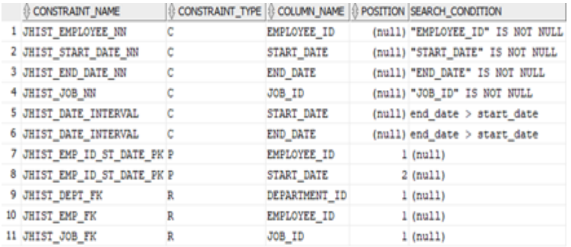
Problem No: 1	**No. Rows:**
Cree las tablas adicionales que se utilizan en esta sección ejecutando las siguientes sentencias:	**2**
Code:	**Result:**

CREATE TABLE emp AS SELECT * FROM employees;
CREATE TABLE dept AS SELECT * FROM departments;
	
	Table EMP created.
	Table DEPT created.

2.

Problem No: 2	**No. Rows**
Cree un informe que muestre el nombre de restricción, el tipo, el nombre de columna y la posición de columna de todas las restricciones de la tabla JOB_HISTORY, además de las restricciones no nulas.	**10**
Code:	**Result:**

```
CREATE TABLE INFORME AS(                                                  SELECT
cons.constraint_name, cons.constraint_type,cols.column_name,
cols.position,TO_LOB(cons.search_condition)search_condition
FROM user_constraints cons
INNER JOIN user_cons_columns cols ON cons.constraint_name = cols.constraint_name
WHERE cons.table_name = 'JOB_HISTORY' );
```

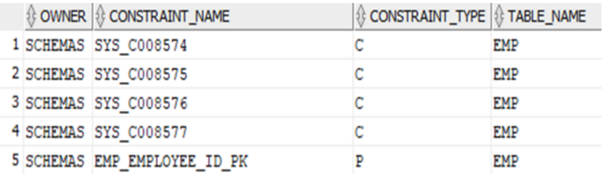


	CONSTRAINT_NAME	CONSTRAINT_TYPE	COLUMN_NAME	POSITION	SEARCH_CONDITION
1	JHIST_EMPLOYEE_NN	C	EMPLOYEE_ID	(null)	"EMPLOYEE_ID" IS NOT NULL
2	JHIST_START_DATE_NN	C	START_DATE	(null)	"START_DATE" IS NOT NULL
3	JHIST_END_DATE_NN	C	END_DATE	(null)	"END_DATE" IS NOT NULL
4	JHIST_JOB_NN	C	JOB_ID	(null)	"JOB_ID" IS NOT NULL
5	JHIST_DATE_INTERVAL	C	START_DATE	(null)	end_date > start_date
6	JHIST_DATE_INTERVAL	C	END_DATE	(null)	end_date > start_date
7	JHIST_EMP_ID_ST_DATE_PK	P	EMPLOYEE_ID	1	(null)
8	JHIST_EMP_ID_ST_DATE_PK	P	START_DATE	2	(null)
9	JHIST_DEPT_FK	R	DEPARTMENT_ID	1	(null)
10	JHIST_EMP_FK	R	EMPLOYEE_ID	1	(null)
11	JHIST_JOB_FK	R	JOB_ID	1	(null)

3.

Problem No: 3	**No. Rows**
Cree una restricción de clave primaria en la columna employee_id de la tabla emp.	**5**
Code:	**Result:**

```
ALTER TABLE emp

ADD CONSTRAINT emp_employee_id_pk PRIMARY KEY (employee_id);
```



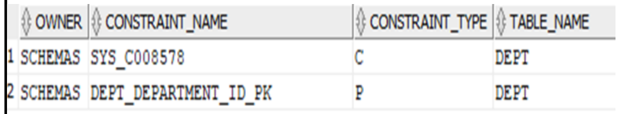
	OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
1	SCHEMAS	SYS_C008574	C	EMP
2	SCHEMAS	SYS_C008575	C	EMP
3	SCHEMAS	SYS_C008576	C	EMP
4	SCHEMAS	SYS_C008577	C	EMP
5	SCHEMAS	EMP_EMPLOYEE_ID_PK	P	EMP

4.

Problem No: 4	No. Rows
Cree una clave primaria en la columna department_id de la tabla dept.	2

Code:	Result:

```
ALTER TABLE dept

ADD CONSTRAINT dept_department_id_pk PRIMARY KEY (department_id);
```



	OWNER	CONSTRAINT_NAME	CONSTRAINT_TYPE	TABLE_NAME
1	SCHEMAS	SYS_C008578	C	DEPT
2	SCHEMAS	DEPT_DEPARTMENT_ID_PK	P	DEPT

5.

Problem No: 5	No. Rows
Agregue una restricción ajena entre DEPT y EMP, de modo que solo se puedan introducir departamentos válidos en la tabla EMP.	1

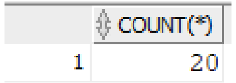
Code:	Result:

`ALTER TABLE emp ADD CONSTRAINT emp_dept_department_id_fk`
`FOREIGN KEY (department_id)REFERENCES dept (department_id) ON DELETE CASCADE;`
	Table EMP altered.

6.

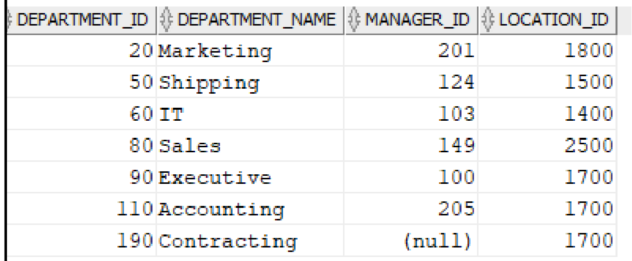
Problem No: 6	No. Rows
Pruebe la restricción de clave ajena que acaba de crear: Cuente el número de filas en la tabla EMP. Elimine el departamento 10 de la tabla dept. Ahora vuelva a contar los empleados. Debería haber menos empleados	1
Code:	Result:

```
select count(*) from emp;


delete from dept where department_id = 10;


select count(*) from emp;
```

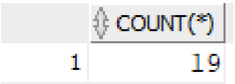


```
  COUNT(*)
1       20
```

```
1 row deleted.
```



DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting	(null)	1700



```
  COUNT(*)
1       19
```

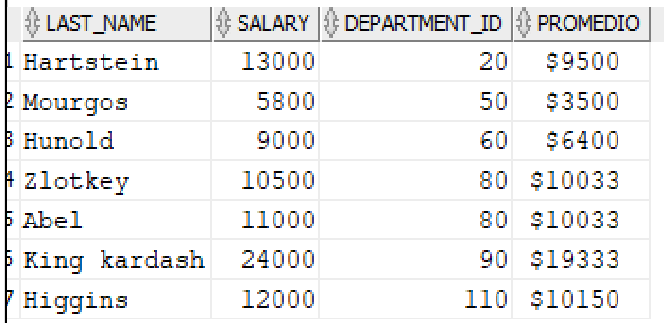
7.

Problem No: 7	No. Rows
Genere un informe que devuelva el apellido, el salario, el número de departamento y El salario medio de todos los departamentos en los que el salario es mayor que el salario medio.	7

Code:

```
with medio as

(select department_id departamento_id, avg(salary) promedio_sal

from employees

group by department_id)

select emp.last_name, emp.salary "SALARY", med.departamento_id "DEPARTMENT_ID",
to_char(med.promedio_sal,'$99999') "PROMEDIO"

from employees emp inner join medio med on emp.department_id = med.departamento_id

where emp.salary > med.promedio_sal

order by med.departamento_id;
```

Result:



	LAST_NAME	SALARY	DEPARTMENT_ID	PROMEDIO
1	Hartstein	13000	20	$9500
2	Mourgos	5800	50	$3500
3	Hunold	9000	60	$6400
4	Zlotkey	10500	80	$10033
5	Abel	11000	80	$10033
6	King kardash	24000	90	$19333
7	Higgins	12000	110	$10150

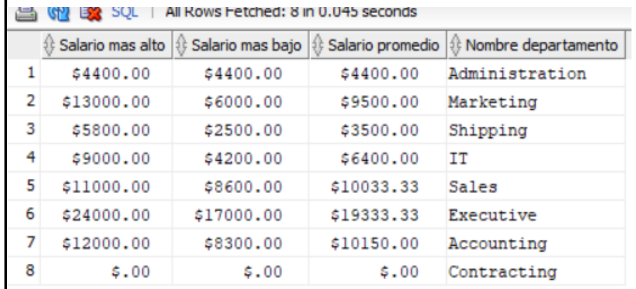
8.

Problem No: 8	**No. Rows**
Cree una vista denominada V2 que devuelva el salario más alto, el salario más bajo, el salario medio y el nombre del departamento.	**8**

Code:

```
CREATE OR REPLACE VIEW V2 ("Salario mas alto", "Salario mas bajo", "Salario promedio",
"Nombre departamento") AS SELECT

  TO_CHAR(ROUND(MAX(NVL(empl.salary,0)),2),'$999999.99'),

  TO_CHAR(ROUND(MIN(NVL(empl.salary,0)),2),'$999999.99'),

  TO_CHAR(ROUND(AVG(NVL(empl.salary,0)),2),'$999999.99'),

  depa.department_name

FROM departments depa LEFT OUTER JOIN employees empl

ON depa.department_id = empl.department_id

GROUP BY (depa.department_id, depa.department_name);
```

Result:



All Rows Fetched: 8 in 0.045 seconds

	Salario mas alto	Salario mas bajo	Salario promedio	Nombre departamento
1	$4400.00	$4400.00	$4400.00	Administration
2	$13000.00	$6000.00	$9500.00	Marketing
3	$5800.00	$2500.00	$3500.00	Shipping
4	$9000.00	$4200.00	$6400.00	IT
5	$11000.00	$8600.00	$10033.33	Sales
6	$24000.00	$17000.00	$19333.33	Executive
7	$12000.00	$8300.00	$10150.00	Accounting
8	$.00	$.00	$.00	Contracting

9.

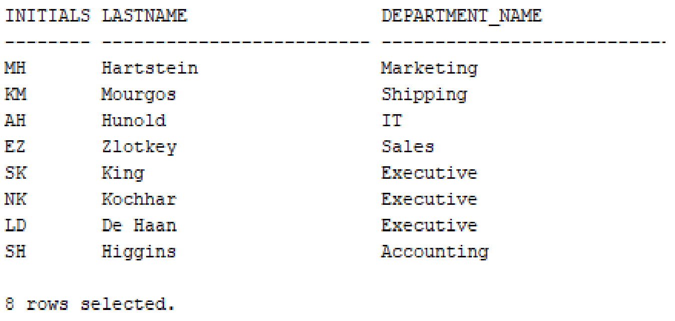
Problem No: 9	**No. Rows**

Cree una vista denominada Dept_Managers_view que devuelva una lista de nombres de departamento junto con las iniciales y el apellido del jefe para dicho departamento. **Pruebe la vista devolviendo todas sus filas. Asegúrese de que no se pueda actualizar ninguna fila a través de la vista.** **Pruebe a ejecutar una sentencia UPDATE en la vista.**	**8**

Code:

```
CREATE OR REPLACE VIEW dept_managers_view

AS SELECT DISTINCT SUBSTR(NVL(mgr.first_name, '_'),1, 1) ||

SUBSTR(mgr.last_name,1, 1) initials, mgr.last_name lastname, dpt.department_name

FROM employees mgr INNER JOIN employees emp

ON mgr.employee_id = emp.manager_id

LEFT OUTER JOIN departments dpt

ON mgr.department_id = dpt.department_id;

SELECT * FROM Dept_Managers_view ;
```

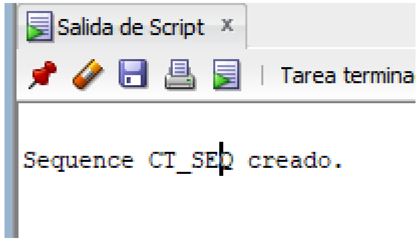
Result:



```
INITIALS LASTNAME                  DEPARTMENT_NAME
-------- ------------------------- ------------------------------
MH       Hartstein                 Marketing
KM       Mourgos                   Shipping
AH       Hunold                    IT
EZ       Zlotkey                   Sales
SK       King                      Executive
NK       Kochhar                   Executive
LD       De Haan                   Executive
SH       Higgins                   Accounting

8 rows selected.
```

10.

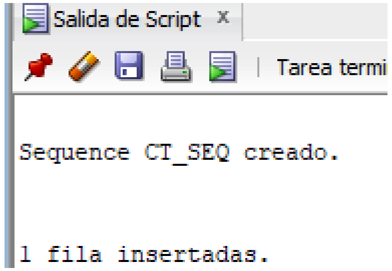
Problem No: 10	**No. Rows**

Cree una secuencia denominada ct_seq con todos los valores por defecto.	**0**
Code:	**Result:**
`CREATE SEQUENCE ct_seq ;`	 Sequence CT_SEQ creado.

11.

Problem No: 11	**No. Rows**
Examine la siguiente sentencia de inserción y corrija los errores.	**0**
Code:	**Result:**

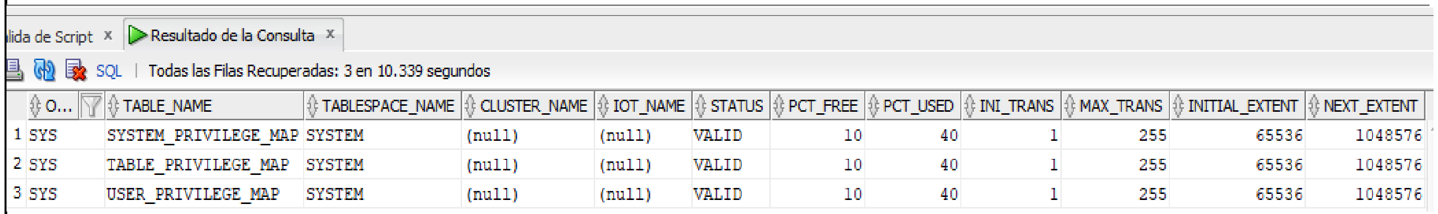
```
INSERT INTO emp

(employee_id, first_name, last_name, email, phone_number,

hire_date,

job_id, salary, commission_pct, manager_id, department_id)

VALUES

(ct_seq.NEXTVAL, 'Kaare', 'Hansen', 'KHANSEN', '44965 832123',

sysdate, 'SA_REP', 6500, null, 100, 20);
```



```
Sequence CT_SEQ creado.


1 fila insertadas.
```

12.

Problem No: 12	**No. Rows**
Escriba la sentencia SQL para mostrar todas las tablas de usuario que contienen el nombre PRIV.	**3**
Code:	**Result:**

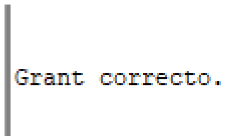
SELECT * FROM all_tables WHERE REGEXP_LIKE(table_name, '(PRIV)');	

Todas las Filas Recuperadas: 3 en 10.339 segundos

	O...	TABLE_NAME	TABLESPACE_NAME	CLUSTER_NAME	IOT_NAME	STATUS	PCT_FREE	PCT_USED	INI_TRANS	MAX_TRANS	INITIAL_EXTENT	NEXT_EXTENT
1	SYS	SYSTEM_PRIVILEGE_MAP	SYSTEM	(null)	(null)	VALID	10	40	1	255	65536	1048576
2	SYS	TABLE_PRIVILEGE_MAP	SYSTEM	(null)	(null)	VALID	10	40	1	255	65536	1048576
3	SYS	USER_PRIVILEGE_MAP	SYSTEM	(null)	(null)	VALID	10	40	1	255	65536	1048576

13.

Problem No: 13	
Conceda acceso de selección a público en la tabla EMP y verifique que se ha otorgado mediante la ejecución esta consulta	**0**
Code:	**Result:**

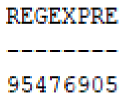
SELECT * FROM user_tab_privs WHERE table_name = 'EMP';
GRANT SELECT ON emp to PUBLIC;
	
	Grant correcto.

14.

Problem No: 14	

Sustituya ?? en la siguiente consulta mediante expresiones regulares para devolver
solo los números de la siguiente cadena: 'Oracle Academy9547d6905%&^ db apex'.
SELECT REGEXP_REPLACE('Oracle Academy9547d6905%&^ db apex',??,'') regexpreplace
FROM DUAL;
	1

Code:	**Result:**

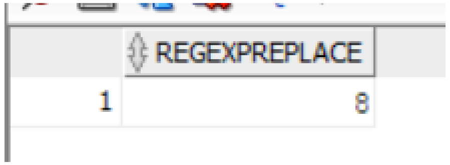
`SELECT REGEXP_REPLACE('Oracle Academy9547d6905%&^ db apex','[^[:digit:]]','') regexpreplace`
`FROM DUAL;`
`SELECT REGEXP_REPLACE('Oracle Academy9547d6905%&^ db apex','[^0-9]','') regexpreplace`
`FROM DUAL;`
	
	`REGEXPRE`
	`--------`
	`95476905`

15.

Problem No: 15	

Corrija la consulta anterior mediante expresiones regulares para devolver el número de
dígitos de la siguiente cadena: 'Oracle Academy9547d6905 %y;^ db'
SELECT LENGTH(REGEXP_REPLACE('Oracle Academy9547d6905%&^ db apex','??','')) regexpreplace
FROM DUAL;
	1

Code:	**Result:**

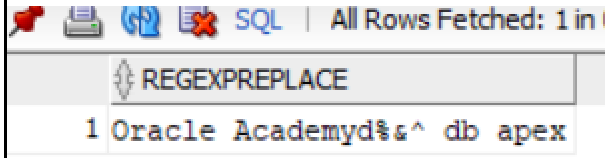
`SELECT LENGTH(REGEXP_REPLACE('Oracle Academy9547d6905%&^ db apex','[^[:digit:]]','')) regexpreplace`
`FROM DUAL;`
	
	REGEXPREPLACE
	1 8

16.

Problem No: 16	

Corrija la consulta de nuevo para devolver sólo los caracteres no numéricos.
SELECT REGEXP_REPLACE('Oracle Academy9547d6905%&^ db apex','??','') regexpreplace
FROM DUAL;
	1

Code:	Result:

`SELECT REGEXP_REPLACE('Oracle Academy9547d6905%&^ db apex','[0-9]','') regexpreplace`
`FROM DUAL;`
	
	
	All Rows Fetched: 1 in
	REGEXPREPLACE
	1 Oracle Academyd%&^ db apex

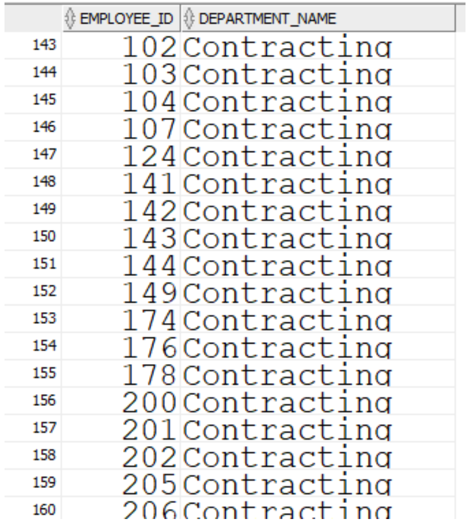
17.

Problem No: 17	
Mediante las uniones propiedad de Oracle, construya una instrucción que devuelva todos los employee_ids unidos a todos los department_names.	**160**

Code:

```
select e.employee_id, d.department_name

from employees e, departments d;
```

Result:



```
    EMPLOYEE_ID DEPARTMENT_NAME
143         102 Contracting
144         103 Contracting
145         104 Contracting
146         107 Contracting
147         124 Contracting
148         141 Contracting
149         142 Contracting
150         143 Contracting
151         144 Contracting
152         149 Contracting
153         174 Contracting
154         176 Contracting
155         178 Contracting
156         200 Contracting
157         201 Contracting
158         202 Contracting
159         205 Contracting
160         206 Contracting
```

18.

Problem No: 18

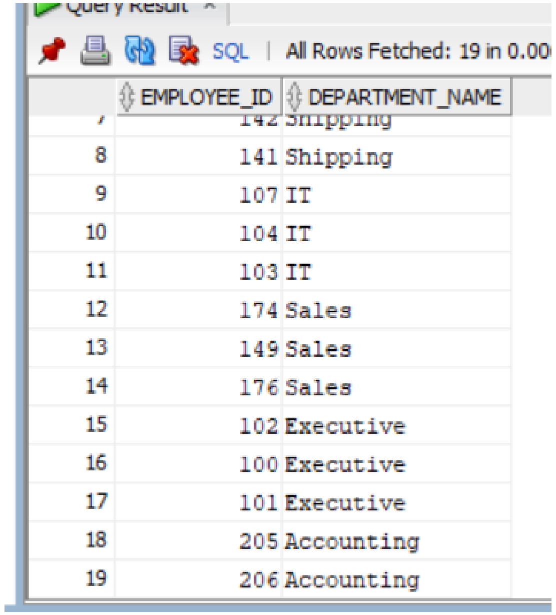
Vuelva a utilizar las uniones Oracle para corregir la sentencia anterior de modo que devuelva solo el nombre del departamento en el que está trabajando el empleado actualmente.

19

Code:

```
select e.employee_id, d.department_name

from employees e, departments d

where e.department_id=d.department_id;
```

Result:



	EMPLOYEE_ID	DEPARTMENT_NAME
7	142	Shipping
8	141	Shipping
9	107	IT
10	104	IT
11	103	IT
12	174	Sales
13	149	Sales
14	176	Sales
15	102	Executive
16	100	Executive
17	101	Executive
18	205	Accounting
19	206	Accounting

19.

Problem No: 19

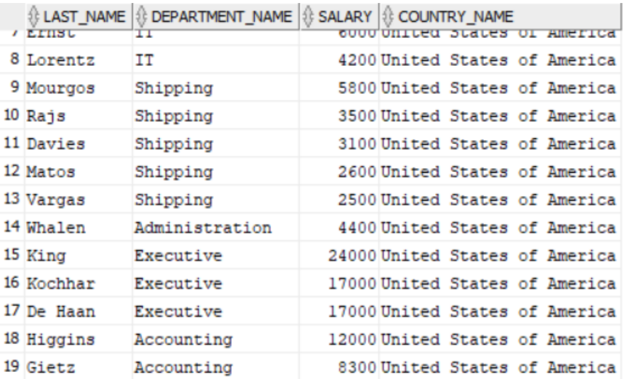
Vuelva a utilizar las uniones Oracle para crear una consulta que muestre el apellido de los empleados, el nombre de departamento, el salario y el nombre del país de todos los empleados.

19

Code:

```
select e.last_name, d.department_name, e.salary, c.country_name

from employees e, departments d, locations loc, countries c

where e.department_id = d.department_id

and d.location_id = loc.location_id(+) and

loc.country_id = c.country_id(+);
```

Result:



	LAST_NAME	DEPARTMENT_NAME	SALARY	COUNTRY_NAME
7	Ernst	IT	6000	United States of America
8	Lorentz	IT	4200	United States of America
9	Mourgos	Shipping	5800	United States of America
10	Rajs	Shipping	3500	United States of America
11	Davies	Shipping	3100	United States of America
12	Matos	Shipping	2600	United States of America
13	Vargas	Shipping	2500	United States of America
14	Whalen	Administration	4400	United States of America
15	King	Executive	24000	United States of America
16	Kochhar	Executive	17000	United States of America
17	De Haan	Executive	17000	United States of America
18	Higgins	Accounting	12000	United States of America
19	Gietz	Accounting	8300	United States of America

20.

Problem No: 20

Vuelva a utilizar la sintaxis de unión de Oracle para modificar la consulta anterior,

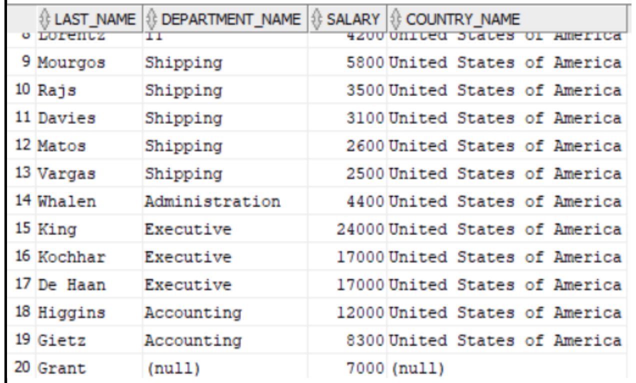
de modo que incluya también incluye el registro de empleado del empleado sin department_id, 'Grant'.

20

Code:

```
select e.last_name, d.department_name, e.salary, c.country_name

from employees e, departments d, locations loc, countries c

where e.department_id = d.department_id(+)

and d.location_id = loc.location_id(+) and

loc.country_id = c.country_id(+);
```

Result:



	LAST_NAME	DEPARTMENT_NAME	SALARY	COUNTRY_NAME
8	Lorentz	IT	4200	United States of America
9	Mourgos	Shipping	5800	United States of America
10	Rajs	Shipping	3500	United States of America
11	Davies	Shipping	3100	United States of America
12	Matos	Shipping	2600	United States of America
13	Vargas	Shipping	2500	United States of America
14	Whalen	Administration	4400	United States of America
15	King	Executive	24000	United States of America
16	Kochhar	Executive	17000	United States of America
17	De Haan	Executive	17000	United States of America
18	Higgins	Accounting	12000	United States of America
19	Gietz	Accounting	8300	United States of America
20	Grant	(null)	7000	(null)