Universidad Politécnica de Aguascalientes

Ingeniería en Sistemas Computacionales

Prácticas Oracle

Administración de Base de Datos

Cristian Alessandro Verdín Mata	UP200220
Eduardo Pulido Guzmán	UP200516
Jaime Jesabel Valadez Láriz	UP200205
Paola Castañeda Serrano	UP200264
César Alejandro Hernández Macías	UP200775

2022-3

Aguascalientes, Aguascalientes
15 de Noviembre de 2022

Index

DP_ 20.1

Exercise 1

Problem No: 1	No. Rows in Result:
Cree las tablas adicionales que se utilizan en esta sección ejecutando las siguientes sentencias.	28

Text Code (No image) :

```
CREATE TABLE emp AS SELECT * FROM employees;
CREATE TABLE dept AS SELECT * FROM departments;
```

Image Result:

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	50	Shipping	124	1500
4	60	IT	103	1400
5	80	Sales	149	2500
6	90	Executive	100	1700
7	110	Accounting	205	1700
8	190	Contracting	(null)	1700

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID	BONUS
1	100	Steven	King	SKING	515.123.4567	17-JUN-87	AD_PRES	24000	(null)	(null)	90	(null)
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21-SEP-89	AD_VP	17000	(null)	100	90	(null)
3	102	Lex	De Haan	LDEHAAN	515.123.4569	13-JAN-93	AD_VP	17000	(null)	100	90	(null)
4	200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-87	AD_ASST	4400	(null)	101	10	(null)
5	205	Shelley	Higgins	SHIGGINS	515.123.8080	07-JUN-94	AC_MGR	12000	(null)	101	110	(null)
6	206	William	Gietz	WGIETZ	515.123.8181	07-JUN-94	AC_ACCOUNT	8300	(null)	205	110	(null)
7	149	Eleni	Zlotkey	EZLOTKEY	011.44.1344.429018	29-JAN-00	SA_MAN	10500	0.2	100	80	1500
8	174	Ellen	Abel	EABEL	011.44.1644.429267	11-MAY-96	SA_REP	11000	0.3	149	80	1700
9	176	Jonathon	Taylor	JTAYLOR	011.44.1644.429265	24-MAR-98	SA_REP	8600	0.2	149	80	1250
10	178	Kimberely	Grant	KGRANT	011.44.1644.429263	24-MAY-99	SA_REP	7000	0.15	149	(null)	(null)

Exercise 2

Problem No: 2	No. Rows in Result:
Cree un informe que muestre el nombre de restricción, el tipo, el nombre de la columna y la posición de la columna de todas las restricciones de la tabla JOB_HISTORY, además de las restricciones no nulas.	11

Text Code (No image) :

```
CREATE table info2 as(
   select c.constraint_name, c.constraint_type,
   a.column_name, a.position,
   to_lob(c.search_condition) as search_condition
   from user_constraints c
   inner join user_cons_columns a
   on c.constraint_name= a.constraint_name
   where c.table_name = 'JOB_HISTORY');
```

Image Result:

	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)

Exercise 3

Problem No: 3	No. Rows in Result:
Cree una restricción de clave primaria en la columna employee_id de la tabla emp	1

Text Code (No image) :

```
alter table emp
    modify employee_id CONSTRAINT Alter_Table_EmployeeID_PP Primary key;
```

Image Result:

```
Table EMP altered.
```

Exercise 4

Problem No: 4	No. Rows in Result:
Cree una clave primaria en la columna department_id de la tabla dept.	1

Text Code (No image) :

```
alter table dept
    modify department_id Primary key;
```

Image Result:

```
Table DEPT altered.
```

Exercise 5

Problem No: 5	No. Rows in Result:
Agregue una restricción ajena entre DEPT y EMP, de modo que solo se puedan introducir departamentos válidos en la tabla EMP. Asegúrese de que puede suprimir cualquier fila de la tabla DEPT y de que se suprimen las filas a las que se hace referencia en la tabla EMP.	1

Text Code (No image) :

```
alter table emp
add constraint Alter_Table_Emp_Dept_Department_Id_FK Foreign key
(department_id) references dept (department_id)
on delete cascade;
```

Image Result:

```
Table EMP altered.
```

Exercise 6

Problem No: 6	No. Rows in Result:
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

Text Code (No image) :

```
select count(*) from emp;
delete from dept where department_id=10;
select count(*) from emp; --volver a contar--
```

Image Result:

	COUNT(*)
1	20

```
1 row deleted.
```

	COUNT(*)
1	19

Exercise 7

Problem No: 7	No. Rows in Result:
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

Text Code (No image) :

```
SELECT e.last_name, e.salary, e.department_id, sa.salavg
FROM emp e, (SELECT department_id, ROUND(AVG(salary),0) salavg
             FROM emp
             GROUP BY department_id) sa
WHERE (e.salary > sa.salavg)
    and (e.department_id = sa.department_id)
ORDER BY e.department_id;
```

Image Result:

	LAST_NAME	SALARY	DEPARTMENT_ID	SALAVG
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	24000	90	19333
7	Higgins	12000	110	10150

Exercise 8

Problem No: 8	No. Rows in Result:
Cree una vista denominada V2 que devuelva el salario más alto, el salario más bajo, el salario medio y el nombre del departamento	7

Text Code (No image) :

```
CREATE VIEW V2 as
SELECT distinct d.department_name, sl.Lowest_Salary, sh.Highest_Salary,
sa.Averege_Salary
FROM dept d, (SELECT department_id, MIN(salary) Lowest_Salary
              FROM emp
              GROUP BY department_id) sl,
              (SELECT department_id, MAX(salary) Highest_Salary
              FROM emp
              GROUP BY department_id) sh,
              (SELECT department_id, ROUND(AVG(salary),0) Averege_Salary
              FROM emp
              GROUP BY department_id) sa, emp e
WHERE (e.department_id = sl.department_id)
    and (e.department_id = sh.department_id)
    and (e.department_id = sa.department_id)
    and (d.department_id = e.department_id)
ORDER BY d.department_name;

SELECT * FROM V2;
```

Image Result:

	DEPARTMENT_NAME	LOWEST_SALARY	HIGHEST_SALARY	AVEREGE_SALARY
1	Accounting	8300	12000	10150
2	Administration	4400	4400	4400
3	Executive	17000	24000	19333
4	IT	4200	9000	6400
5	Marketing	6000	13000	9500
6	Sales	8600	11000	10033
7	Shipping	2500	5800	3500

Exercise 9

Problem No: 9	No. Rows in Result:
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. Asegures que no se pueda actualizar ninguna gila a través de la vista. Pruebe a ejecutar una sentencia UPDATE en la vista.	6

Text Code (No image) :

```
create VIEW Dept_Managers_View as (
 select d.department_name,SUBSTR(a.first_name,1,1)||' ' ||a.last_name as
name from DEPARTMENTS d,
          (select employee_id, first_name ,last_name
           from employees
           where employee_id in (select DISTINCT(manager_id)
                                 from employees)
           order by employee_id) a
where d.manager_id = a.employee_id)
with read only;
```

Image Result:

	DEPARTMENT_NAME	NAME
1	Executive	S King
2	IT	A Hunold
3	Shipping	K Mourgos
4	Sales	E Zlotkey
5	Marketing	M Hartstein
6	Accounting	S Higgins

Exercise 10

Problem No: 10	No. Rows in Result:
Cree una secuencia denominada ct_seq con todos los valores por defecto.	1
Text Code (No image) :	
`CREATE SEQUENCE ct_seq;`	
Image Result:	
Sequence CT_SEQ created.	

Exercise 11

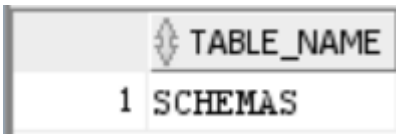
Problem No: 11	No. Rows in Result:
Examine la siguiente sentencia de inserción y corrija los errores. INSERT INTO emp (employee_id, first_name, last_name, email, phone_number, hire_date, job_id, salary, commission_pct, manager_id, department_id) VALUES (ct_seq.nextvalue,”Kaare”,’Hansen’,’KHANSEN’,’44965 832123’, sysdate, ‘SA_REP’, $6500, null, 100, 20);	1
Text Code (No image) :	

```
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', '449.658.32123',sysdate,
    'SA_REP', 6500, null, 100, 20);
```

Image Result:

20	1	Kaare	Hansen	KHANSEN	449.658.32123	14-NOV-22	SA_REP	6500	(null)	100	20	(null)

Exercise 12

Problem No: 12	No. Rows in Result:
Escriba la sentencia SQL para mostrar todas las tablas de usuario que contienen el nombre de PRIV.	1
Text Code (No image) :	
`SELECT table_name from user_tab_privs;`	
Image Result:	
	

Exercise 13

Problem No: 13	No. Rows in Result:
Conceda acceso de selección a público en la tabla EMP y verifique que se ha otorgado mediante la ejecución de esta consulta.	
Text Code (No image) :	

`SELECT *`
`FROM user_tab_privs`
`WHERE table_name = 'EMP';`	

Image Result:	

Exercise 14

Problem No: 14	No. Rows in Result:

Conceda acceso de selección a público en la tabla EMP y verifique que se ha otorgado mediante la ejecución esta consulta. SELECT * FROM user_tab_privs WHERE table_name = 'EMP'; 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

Text Code (No image) :

```
SELECT REGEXP_REPLACE('Oracle Academy9547d6905%&^ db
apex','[^[:digit:]]','')
regexpreplace from DUAL;
```

Image Result:

	REGEXPREPLACE
1	95476905

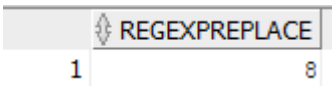
Exercise 15

Problem No: 15	No. Rows in Result:
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

Text Code (No image) :

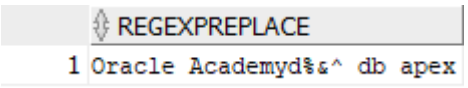
```
SELECT LENGTH(REGEXP_REPLACE('Oracle Academy9547d6905 %y;^
db','[^[:digit:]]','''))
regexpreplace FROM DUAL;
```

Image Result:



	REGEXPREPLACE
1	8

Exercise 16

Problem No: 16	No. Rows in Result:
Corrija la consulta de nuevo para devolver solo los caracteres no numéricos.	1
Text Code (No image) :	
`SELECT REGEXP_REPLACE('Oracle Academy9547d6905%&^ db apex','[0-9]','') regexpreplace from DUAL;`	
Image Result:	
	

Exercise 17

Problem No: 17	No. Rows in Result:
Mediante las uniones propiedad de Oracle, construya una instrucción que devuelva todos los employee_ids unidos a todos los department_names.	16
Text Code (No image) :	

`select  e.employee_id, d.department_name`
`from employees e, departments d`
`order by employee_id;`	

Image Result:	

	EMPLOYEE_ID	DEPARTMENT_NAME
1	100	Administration
2	100	Marketing
3	100	Shipping
4	100	IT
5	100	Sales
6	100	Executive
7	100	Accounting
8	100	Contracting
9	101	Contracting
10	101	Sales
11	101	Administration
12	101	Marketing
13	101	IT
14	101	Shipping
15	101	Accounting
16	101	Executive

Exercise 18

Problem No: 18	**No. Rows in Result:**
Vuelva a utilizar las uniones Oracle para corregir la sentencia anterior de modo que devuelva sólo el nombre del departamento en el que está trabajando el empleado actualmente.	16
Text Code (No image) :	

`select  e.employee_id, d.department_name`
`from employees e, departments d`
`where e.department_id = d.department_id`
`order by employee_id;`	

Image Result:	

	EMPLOYEE_ID	DEPARTMENT_NAME
1	100	Executive
2	101	Executive
3	102	Executive
4	103	IT
5	104	IT
6	107	IT
7	124	Shipping
8	141	Shipping
9	142	Shipping
10	143	Shipping
11	144	Shipping
12	149	Sales
13	174	Sales
14	176	Sales
15	200	Administration
16	201	Marketing

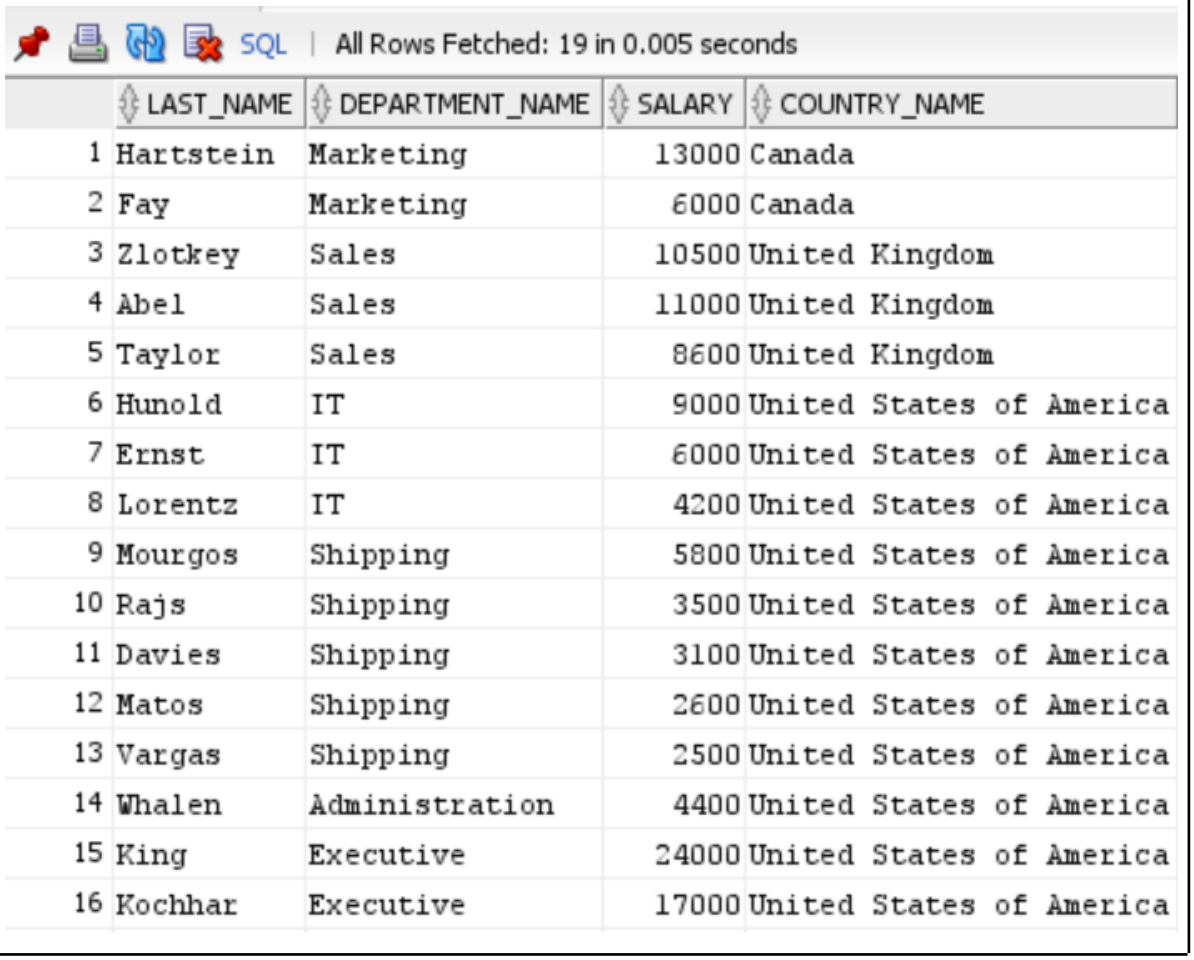
Exercise 19

Problem No: 19	**No. Rows in Result:**
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.	16
Text Code (No image) :	

```
SELECT e.last_name , d.department_name ,e.salary, c.country_name
FROM employees e, departments d, locations l, countries c
WHERE e.department_id = d.department_id
AND
```

```
d.location_id = l.location_id(+)
AND
l.country_id = c.country_id(+)
;
```

Image Result:



SQL | All Rows Fetched: 19 in 0.005 seconds

	LAST_NAME	DEPARTMENT_NAME	SALARY	COUNTRY_NAME
1	Hartstein	Marketing	13000	Canada
2	Fay	Marketing	6000	Canada
3	Zlotkey	Sales	10500	United Kingdom
4	Abel	Sales	11000	United Kingdom
5	Taylor	Sales	8600	United Kingdom
6	Hunold	IT	9000	United States of America
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

Exercise 20

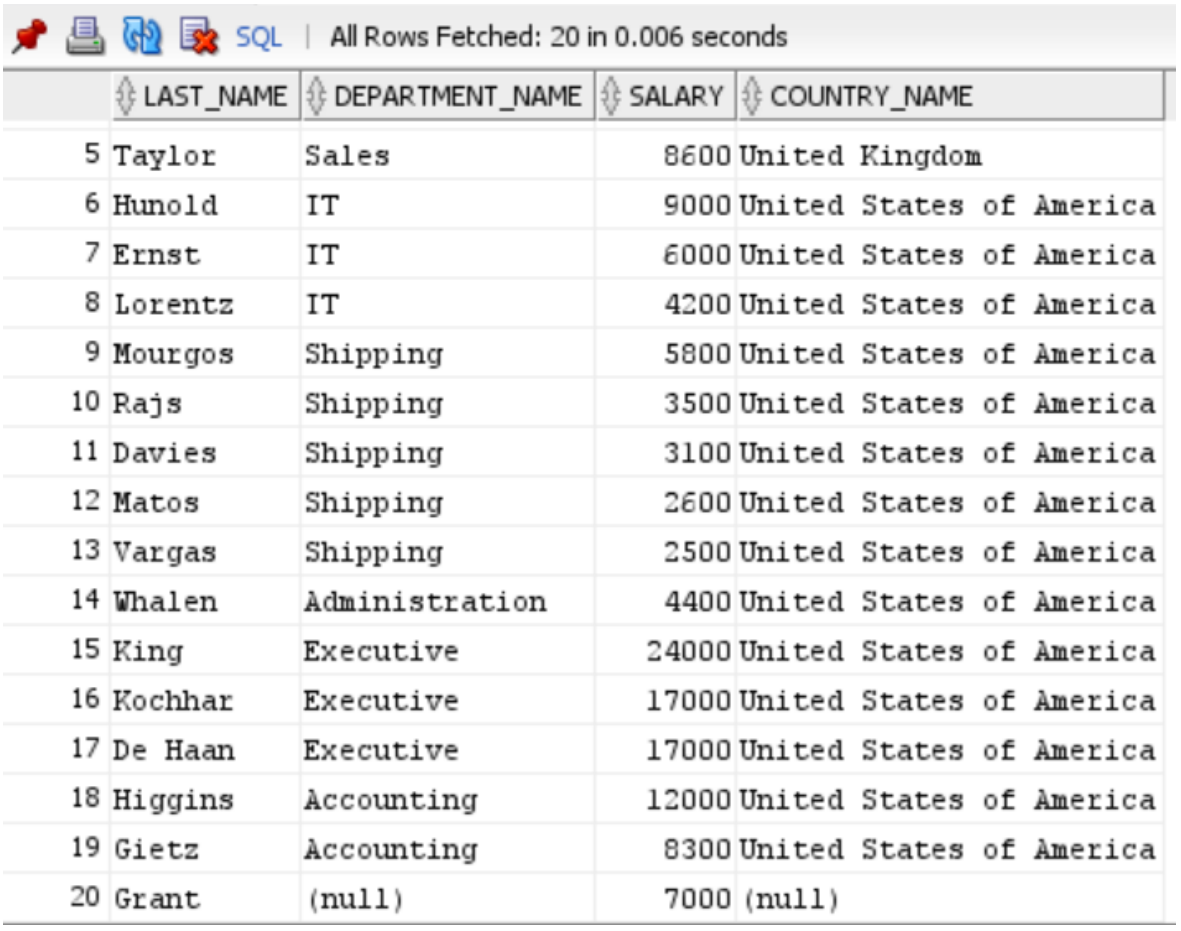
Problem No: 20	No. Rows in Result:
Vuelva a utilizar la sintaxis de unión de Oracle para modificar la consulta anterior, de modo que incluya también incluye el registro de empleado del empleado sin department_id, 'Grant'.	20

Text Code (No image) :

```
SELECT e.last_name , d.department_name ,e.salary, c.country_name
FROM employees e, departments d, locations l, countries c
```

```
WHERE e.department_id = d.department_id(+)
AND
d.location_id = l.location_id(+)
AND
l.country_id = c.country_id(+)
;
```

Image Result:



All Rows Fetched: 20 in 0.006 seconds

	LAST_NAME	DEPARTMENT_NAME	SALARY	COUNTRY_NAME
5	Taylor	Sales	8600	United Kingdom
6	Hunold	IT	9000	United States of America
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	Grant	(null)	7000	(null)