

Nombre:

Isaac Rafael

Apellidos:

Fermín Llauger

MATRÍCULA:

2024-0397

TEMA:

Practica SQL Developer

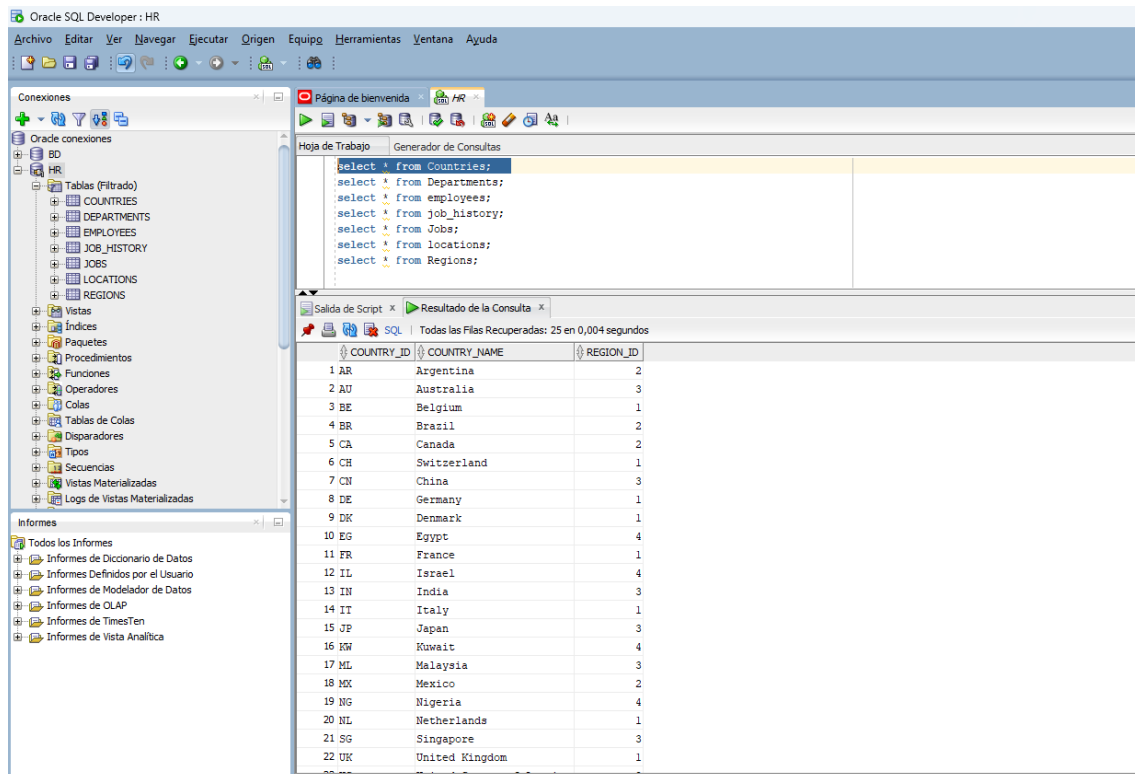
Profesor:

José Manuel Aquino



1. Mostrar el contenido de cada una de las tablas del esquema de hr.

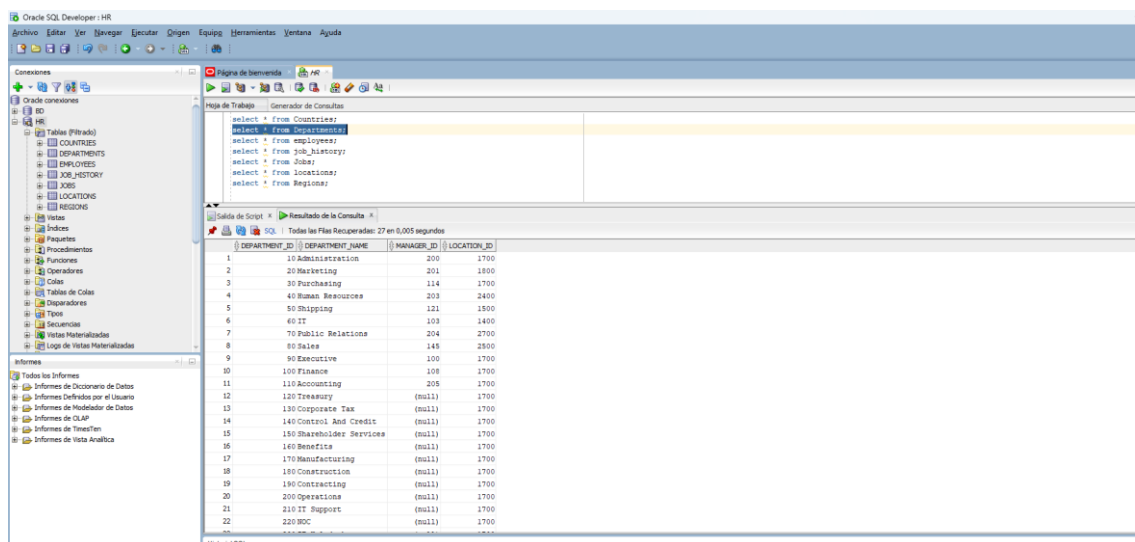
Tabla Countries



The screenshot shows the Oracle SQL Developer interface. The left pane displays the 'Conexiones' (Connections) tree with the 'HR' schema selected. The main window shows the 'Hoja de Trabajo' (Worksheet) with a SQL query: `select * from Countries;`. The 'Resultado de la Consulta' (Query Result) pane shows the data for the Countries table, with 25 rows and 3 columns: COUNTRY_ID, COUNTRY_NAME, and REGION_ID.

COUNTRY_ID	COUNTRY_NAME	REGION_ID
1 AR	Argentina	2
2 AU	Australia	3
3 BE	Belgium	1
4 BR	Brazil	2
5 CA	Canada	2
6 CH	Switzerland	1
7 CN	China	3
8 DE	Germany	1
9 DK	Denmark	1
10 EG	Egypt	4
11 FR	France	1
12 IL	Israel	4
13 IN	India	3
14 IT	Italy	1
15 JP	Japan	3
16 KW	Kuwait	4
17 ML	Malaysia	3
18 MX	Mexico	2
19 NG	Nigeria	4
20 NL	Netherlands	1
21 SG	Singapore	3
22 UK	United Kingdom	1

Tabla Departments



The screenshot shows the Oracle SQL Developer interface. The left pane displays the 'Conexiones' (Connections) tree with the 'HR' schema selected. The main window shows the 'Hoja de Trabajo' (Worksheet) with a SQL query: `select * from Departments;`. The 'Resultado de la Consulta' (Query Result) pane shows the data for the Departments table, with 27 rows and 4 columns: DEPARTMENT_ID, DEPARTMENT_NAME, MANAGER_ID, and LOCATION_ID.

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10 Administration	200	1700
2	20 Marketing	201	1800
3	30 Purchasing	114	1700
4	40 Human Resources	203	2400
5	50 Shipping	121	1500
6	60 IT	103	1400
7	70 Public Relations	204	2700
8	80 Sales	145	2500
9	90 Executive	100	1700
10	100 Finance	108	1700
11	110 Accounting	205	1700
12	120 Treasury	(null)	1700
13	130 Corporate Tax	(null)	1700
14	140 Control And Credit	(null)	1700
15	150 Shareholder Services	(null)	1700
16	160 Benefits	(null)	1700
17	170 Manufacturing	(null)	1700
18	180 Contracting	(null)	1700
19	190 Contracting	(null)	1700
20	200 Operations	(null)	1700
21	210 IT Support	(null)	1700
22	220 BOC	(null)	1700

Tabla Employees

Oracle SQL Developer - HR

Archivo Editar Ver Navegar Ejecutar Origen Equipo Herramientas Ventana Ayuda

Conexiones

Oracle conexiones

BD

HR

Tablas (Filtrado)

COUNTRIES

DEPARTMENTS

EMPLOYEES

JOB_HISTORY

JOBS

LOCATIONS

REGIONS

Vistas

Indices

Paquetes

Procedimientos

Funciones

Operadores

Colas

Tablas de Colas

Disparadores

Tipos

Secuencias

Vistas Materializadas

Log de Vistas Materializadas

Informes

Todos los Informes

Informes de Diccionario de Datos

Informes Definidos por el Usuario

Informes de Modelador de Datos

Informes de OLAP

Informes de TimesTen

Informes de Vista Analítica

Página de bienvenida

HR

Hoja de Trabajo

Generador de Consultas

select * from Countries;
select * from Departments;
select * from employees;
select * from job_history;
select * from Jobs;
select * from Locations;
select * from Regions;

Salida de Script x Resultado de la Consulta x

Se han recuperado 50 filas en 0,013 segundos

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	100	Steven	King	SKING	515.123.4567	17/06/03	AD_PRES	24000	(null)	(null)	90
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21/09/05	AD_VP	17000	(null)	100	90
3	102	Lex	De Haan	LDEHAAN	515.123.4569	13/01/01	AD_VP	17000	(null)	100	90
4	103	Alexander	Rumold	ARUMOLD	590.423.4567	03/01/06	IT_PROG	9000	(null)	102	60
5	104	Bruce	Ernst	BERNST	590.423.4568	21/05/07	IT_PROG	6000	(null)	103	60
6	105	David	Austin	DAUSTIN	590.423.4569	25/06/05	IT_PROG	4800	(null)	103	60
7	106	Valli	Pataballa	VPATABAL	590.423.4560	05/02/06	IT_PROG	4800	(null)	103	60
8	107	Diana	Lorentz	DLARENTZ	590.423.5567	07/02/07	IT_PROG	4200	(null)	103	60
9	108	Nancy	Greenberg	NGREENBE	515.124.4569	17/08/02	FI_MGR	12008	(null)	101	100
10	109	Daniel	Faviet	DFAVIET	515.124.4169	16/08/02	FI_ACCOUNT	9000	(null)	108	100
11	110	John	Chen	JCHEN	515.124.4269	28/09/05	FI_ACCOUNT	8200	(null)	108	100
12	111	Ismail	Sciarra	ISCIARRA	515.124.4369	30/09/05	FI_ACCOUNT	7700	(null)	108	100
13	112	Jose Manuel	Urman	JURMAN	515.124.4469	07/03/06	FI_ACCOUNT	7800	(null)	108	100
14	113	Luis	Popp	LPOPF	515.124.4567	07/12/07	FI_ACCOUNT	6900	(null)	108	100
15	114	Den	Raphaely	DRAPHEAL	515.127.4561	07/12/02	PO_MAN	11000	(null)	100	30
16	115	Alexander	Koo	AKOO	515.127.4562	18/05/03	PO_CLERK	3100	(null)	114	30
17	116	Shelli	Beale	SBAIDA	515.127.4563	24/12/05	PO_CLERK	2900	(null)	114	30
18	117	Sigal	Tobias	STOBIAS	515.127.4564	24/07/05	PO_CLERK	2800	(null)	114	30
19	118	Guy	Himuro	GHIIMURO	515.127.4565	15/11/06	PO_CLERK	2600	(null)	114	30
20	119	Karen	Colmenares	KCOLMENAR	515.127.4566	10/09/07	PO_CLERK	2500	(null)	114	30
21	120	Matthew	Weiss	MWEISS	650.123.1234	18/07/04	ST_MAN	8000	(null)	100	50
22	121	Adam	Fripp	AFRIPP	650.123.2234	10/04/05	ST_MAN	8200	(null)	100	50

Tabla Job_History

Oracle SQL Developer - HR

Archivo Editar Ver Navegar Ejecutar Origen Equipo Herramientas Ventana Ayuda

Conexiones

Oracle conexiones

BD

HR

Tablas (Filtrado)

COUNTRIES

DEPARTMENTS

EMPLOYEES

JOB_HISTORY

JOBS

LOCATIONS

REGIONS

Vistas

Indices

Paquetes

Procedimientos

Funciones

Operadores

Colas

Tablas de Colas

Disparadores

Tipos

Secuencias

Vistas Materializadas

Log de Vistas Materializadas

Informes

Todos los Informes

Informes de Diccionario de Datos

Informes Definidos por el Usuario

Informes de Modelador de Datos

Informes de OLAP

Informes de TimesTen

Informes de Vista Analítica

Página de bienvenida

HR

Hoja de Trabajo

Generador de Consultas

select * from Countries;
select * from Departments;
select * from employees;
select * from job_history;
select * from Jobs;
select * from Locations;
select * from Regions;

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 10 en 0,012 segundos

	EMPLOYEE_ID	START_DATE	END_DATE	JOB_ID	DEPARTMENT_ID
1	102	13/01/01	24/07/04	IT_PROG	60
2	101	21/09/97	27/10/01	AC_ACCOUNT	110
3	101	28/10/01	15/03/05	AC_MGR	110
4	201	17/02/04	19/12/07	MSL_REP	20
5	114	24/03/04	31/12/07	ST_CLERK	50
6	122	01/01/07	31/12/07	ST_CLERK	50
7	200	17/09/94	17/06/01	AD_ASST	90
8	174	24/03/04	31/12/04	SA_REP	80
9	174	01/01/07	31/12/07	SA_MAN	80
10	200	01/07/02	31/12/04	AC_ACCOUNT	90

Tabla Jobs

Oracle SQL Developer: HR

ArchivoEditarVerNavegarEjecutarOrigenEquiposHerramientasVentanaAyuda

Conexiones

Oracle conexiones

SD

HR

Tablas (Filtrado)

COUNTRIES

DEPARTMENTS

EMPLOYEES

JOB_HISTORY

JOBS

LOCATIONS

REGIONS

Vistas

Indices

Paquetes

Procedimientos

Funciones

Operadores

Colas

Tablas de Colas

Disparadores

Tipos

Secuencias

Vistas Materializadas

Log de Vistas Materializadas

Informes

Todos los Informes

Informes de Diccionario de Datos

Informes Definidos por el Usuario

Informes de Modelador de Datos

Informes de OLAP

Informes de TimesTen

Informes de Vista Analítica

Página de bienvenida

HR

Hoja de Trabajo

Generador de Consultas

```
select * from Countries;
select * from Departments;
select * from employees;
select * from job_history;
select * from Jobs;
select * from locations;
select * from Regions;
```

Salida de ScriptResultado de la Consulta

Todas las Filas Recuperadas: 19 en 0,006 segundos

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
1 AD_PRES	President	20080	40000
2 AD_VP	Administration Vice President	15000	30000
3 AD_ASST	Administration Assistant	3000	6000
4 FI_MGR	Finance Manager	8200	16000
5 FI_ACCOUNT	Accountant	4200	9000
6 AC_MGR	Accounting Manager	8200	16000
7 AC_ACCOUNT	Public Accountant	4200	9000
8 SA_MGR	Sales Manager	10000	20080
9 SA_REP	Sales Representative	6000	12008
10 PU_MGR	Purchasing Manager	8000	15000
11 PU_CLERK	Purchasing Clerk	2500	5500
12 ST_MGR	Stock Manager	5500	8500
13 ST_CLERK	Stock Clerk	2008	5000
14 SH_CLERK	Shipping Clerk	2500	5500
15 IT_PROG	Programmer	4000	10000
16 MK_MGR	Marketing Manager	9000	15000
17 MK_REP	Marketing Representative	4000	9000
18 HR_REP	Human Resources Representative	4000	9000
19 PR_REP	Public Relations Representative	4500	10500

Tabla Locations

Oracle SQL Developer: HR

ArchivoEditarVerNavegarEjecutarOrigenEquiposHerramientasVentanaAyuda

Conexiones

Oracle conexiones

SD

HR

Tablas (Filtrado)

COUNTRIES

DEPARTMENTS

EMPLOYEES

JOB_HISTORY

JOBS

LOCATIONS

REGIONS

Vistas

Indices

Paquetes

Procedimientos

Funciones

Operadores

Colas

Tablas de Colas

Disparadores

Tipos

Secuencias

Vistas Materializadas

Log de Vistas Materializadas

Informes

Todos los Informes

Informes de Diccionario de Datos

Informes Definidos por el Usuario

Informes de Modelador de Datos

Informes de OLAP

Informes de TimesTen

Informes de Vista Analítica

Página de bienvenida

HR

Hoja de Trabajo

Generador de Consultas

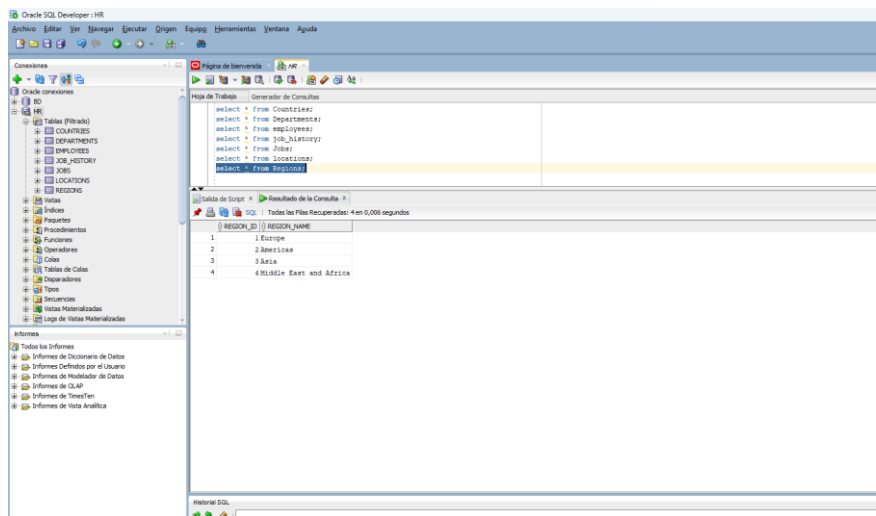
```
select * from Countries;
select * from Departments;
select * from employees;
select * from job_history;
select * from Jobs;
select * from locations;
select * from Regions;
```

Salida de ScriptResultado de la Consulta

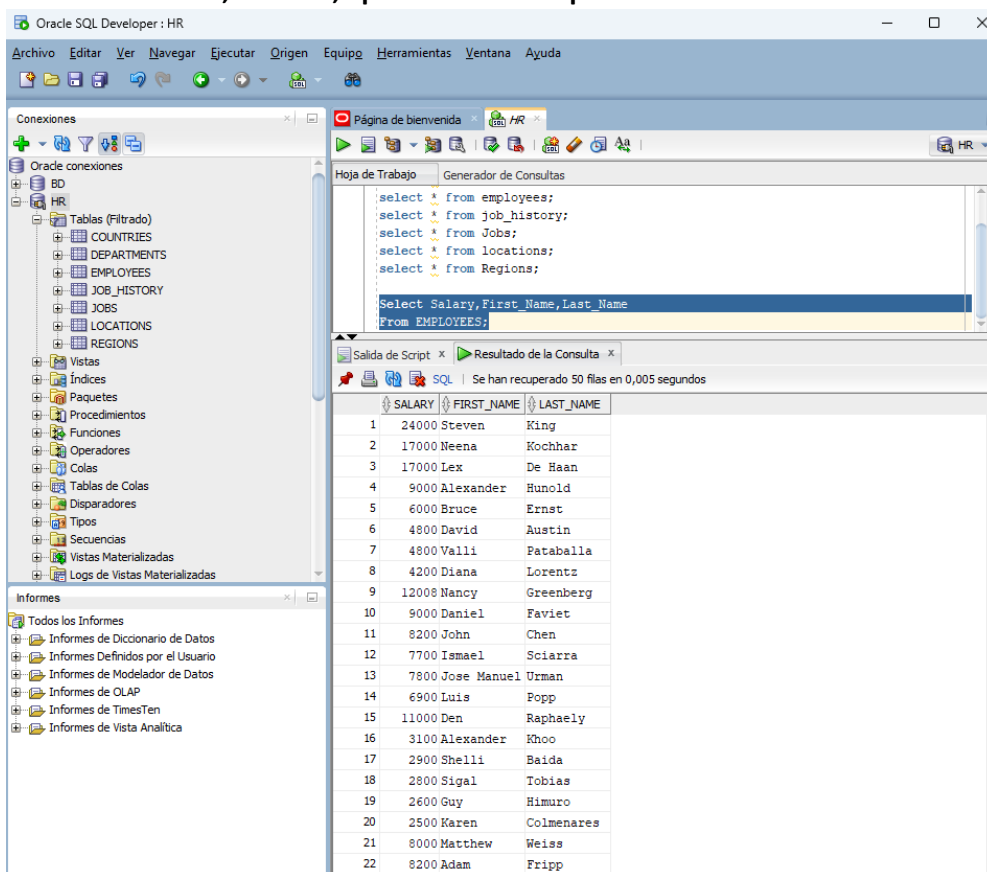
Todas las Filas Recuperadas: 23 en 0,007 segundos

LOCATION_ID	STREET_ADDRESS	POSTAL_CODE	CITY	STATE_PROVINCE	COUNTRY_ID
1	1000 1297 Via Cola di Rie	00989	Roma	(null)	IT
2	1100 93091 Calle della Testa	10934	Venice	(null)	IT
3	1200 2017 Shinjuku-ku	1689	Tokyo	Tokyo Prefecture	JP
4	1300 9450 Kamiya-cho	6623	Hiroshima	(null)	JP
5	1400 2014 Jabberwocky Rd	24192	Southlake	Texas	US
6	1500 2011 Interior Bldv	95034	South San Francisco	California	US
7	1600 2007 Zagora St	50090	South Brunswick	New Jersey	US
8	1700 2004 Charade Rd	98199	Seattle	Washington	US
9	1800 147 Spadina Ave	M5V 2L7	Toronto	Ontario	CA
10	1900 6092 Bonwood St	Y5W 9T2	Whitehorse	Yukon	CA
11	2000 40-5-12 Laogianggen	190518	Beijing	(null)	CN
12	2100 1298 Vileparle (E)	490231	Bombay	Maharashtra	IN
13	2200 12-98 Victoria Street	2901	Sydney	New South Wales	AU
14	2300 190 Clementi North	540198	Singapore	(null)	SG
15	2400 8204 Arthur St	(null)	London	(null)	UK
16	2500 Magdalen Centre, The Oxford Science Park	OX9 9EB	Oxford	Oxford	UK
17	2600 9702 Chester Road	09629650293	Stretford	Manchester	UK
18	2700 Schwannhauserstr. 7031	80925	Munich	Bavaria	DE
19	2800 Rua Frei Caneca 1360	01307-002	Sao Paulo	Sao Paulo	BR
20	2900 20 Rue des Corps-Saints	1730	Geneva	Geneve	CH
21	3000 Murtensstrasse 921	3095	Bern	BE	CH
22	3100 Pieter Breughelstraat 837	3025SK	Utrecht	Utrecht	NL

Tabla Regions



2. Mostrar el salario, nombre, apellido de los empleados.



3. Presentar el código del empleado, su nombre y apellido y su nombre concatenado con una coma espacio y luego el apellido.

En este caso, hice la concatenación con la función CONCAT

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema for the 'HR' schema, including tables like COUNTRIES, DEPARTMENTS, EMPLOYEES, and EMPLOYEE_ID. The main window shows a SQL query in the 'Hoja de Trabajo' (Worksheet) tab:

```
First_name || ', ' || last_name AS Nombre_Concatenado
From employees;

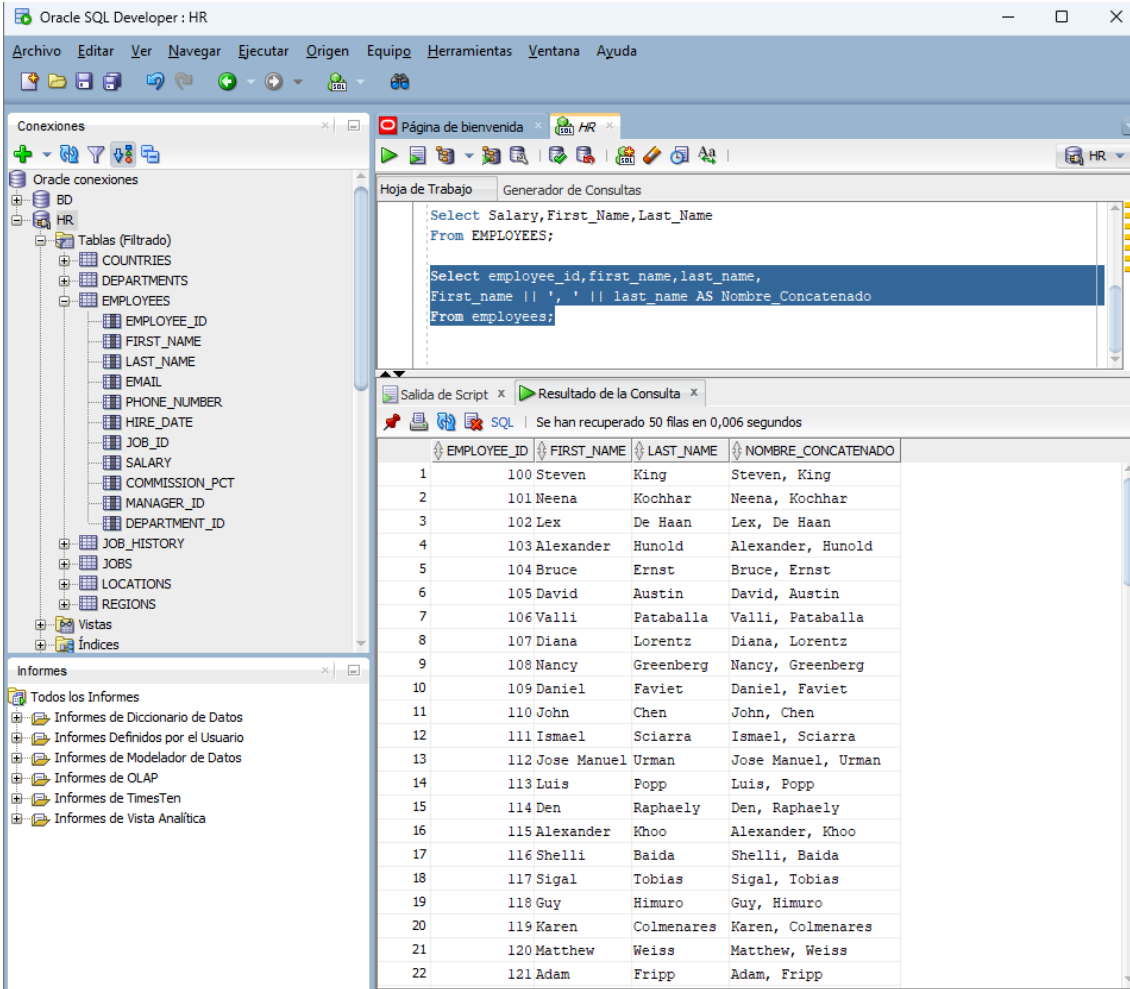
Select employee_id,first_name,last_name,
CONCAT(first_name,CONCAT(', ', last_name)) AS Nombre_Concatenado
From employees;
```

The 'Resultado de la Consulta' (Query Result) tab shows the results of the query. It displays a table with 22 rows and 4 columns: EMPLOYEE_ID, FIRST_NAME, LAST_NAME, and NOMBRE_CONCATENADO. The data is as follows:

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	NOMBRE_CONCATENADO
1	100	Steven King	Steven, King
2	101	Neena Kochhar	Neena, Kochhar
3	102	Lex De Haan	Lex, De Haan
4	103	Alexander Hunold	Alexander, Hunold
5	104	Bruce Ernst	Bruce, Ernst
6	105	David Austin	David, Austin
7	106	Valli Pataballa	Valli, Pataballa
8	107	Diana Lorentz	Diana, Lorentz
9	108	Nancy Greenberg	Nancy, Greenberg
10	109	Daniel Faviet	Daniel, Faviet
11	110	John Chen	John, Chen
12	111	Ismael Sciarra	Ismael, Sciarra
13	112	Jose Manuel Urman	Jose Manuel, Urman
14	113	Luis Popp	Luis, Popp
15	114	Den Raphaely	Den, Raphaely
16	115	Alexander Khoo	Alexander, Khoo
17	116	Shelli Baida	Shelli, Baida
18	117	Sigal Tobias	Sigal, Tobias
19	118	Guy Himuro	Guy, Himuro
20	119	Karen Colmenares	Karen, Colmenares
21	120	Matthew Weiss	Matthew, Weiss
22	121	Adam Fripp	Adam, Fripp

The bottom pane shows the 'Historial SQL' (SQL History) tab, which lists the executed SQL statements and their execution details.

4. En este caso, hice la concatenación con el operador ||



Oracle SQL Developer : HR

Archivo Editar Ver Navegar Ejecutar Origen Equipo Herramientas Ventana Ayuda

Conexiones

Oracle conexiones

BD

HR

Tablas (Filtrado)

COUNTRIES

DEPARTMENTS

EMPLOYEES

EMPLOYEE_ID

FIRST_NAME

LAST_NAME

EMAIL

PHONE_NUMBER

HIRE_DATE

JOB_ID

SALARY

COMMISSION_PCT

MANAGER_ID

DEPARTMENT_ID

JOB_HISTORY

JOBS

LOCATIONS

REGIONS

Vistas

Índices

Informes

Todos los Informes

Informes de Diccionario de Datos

Informes Definidos por el Usuario

Informes de Modelador de Datos

Informes de OLAP

Informes de TimesTen

Informes de Vista Analítica

Página de bienvenida

HR

Hoja de Trabajo

Generador de Consultas

```
Select Salary,First_Name,Last_Name
From EMPLOYEES;

Select employee_id,first_name,last_name,
First_name || ', ' || last_name AS Nombre_Concatenado
From employees;
```

Salida de Script x Resultado de la Consulta x

SQL Se han recuperado 50 filas en 0,006 segundos

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	NOMBRE_CONCATENADO
1	100	Steven	King	Steven, King
2	101	Meena	Kochhar	Meena, Kochhar
3	102	Lex	De Haan	Lex, De Haan
4	103	Alexander	Hunold	Alexander, Hunold
5	104	Bruce	Ernst	Bruce, Ernst
6	105	David	Austin	David, Austin
7	106	Valli	Pataballa	Valli, Pataballa
8	107	Diana	Lorentz	Diana, Lorentz
9	108	Nancy	Greenberg	Nancy, Greenberg
10	109	Daniel	Faviet	Daniel, Faviet
11	110	John	Chen	John, Chen
12	111	Ismael	Sciarra	Ismael, Sciarra
13	112	Jose Manuel	Urman	Jose Manuel, Urman
14	113	Luis	Popp	Luis, Popp
15	114	Den	Raphaely	Den, Raphaely
16	115	Alexander	Khoo	Alexander, Khoo
17	116	Shelli	Baida	Shelli, Baida
18	117	Sigal	Tobias	Sigal, Tobias
19	118	Guy	Himuro	Guy, Himuro
20	119	Karen	Colmenares	Karen, Colmenares
21	120	Matthew	Weiss	Matthew, Weiss
22	121	Adam	Frapp	Adam, Frapp

- Mostrar el código, salario y nombre del empleado. Adicionalmente Mostrar una proyección de la comisión. Esto se compone del salario por la comisión. También realizar un cálculo de proyección con la comisión aumentada en un 10%. Los empleados sin comisión deben presentar 0 en las proyecciones.

En este caso utilice una sentencia CASE para asignarle 0 a los valores NULL

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Conexiones' pane shows the 'HR' schema with tables like EMPLOYEES, DEPARTMENTS, and EMPLOYER_ID. The main window displays a SQL query in the 'Hoja de Trabajo' tab, which is a SELECT statement using CASE WHEN to calculate projections for commission percentage. The 'Resultado de la Consulta' pane shows the results of the query, displaying columns for EMPLOYEE_ID, SALARY, FIRST_NAME, PROYECCION, and PROYECCION10. The results show 100 rows of data, including employee details and their calculated projections.

```
SELECT Employee_ID, salary, first_name,
CASE WHEN commission_pct IS NULL
THEN 0
ELSE
Salary * commission_pct
END AS Proyeccion,
CASE WHEN commission_pct IS NULL
THEN 0
ELSE
(commission_pct * 0.10 + commission_pct) * Salary
END AS Proyeccion10
FROM employees;
```

EMPLOYEE_ID	SALARY	FIRST_NAME	PROYECCION	PROYECCION10
37	136	2200 Hazel	0	0
38	137	3600 Renske	0	0
39	138	3200 Stephen	0	0
40	139	2700 John	0	0
41	140	2500 Joshua	0	0
42	141	3500 Tenna	0	0
43	142	3100 Curtis	0	0
44	143	2600 Randall	0	0
45	144	2500 Peter	0	0
46	145	14000 John	5600	6160
47	146	13500 Karen	4050	4455
48	147	12000 Alberto	3600	3960
49	148	11000 Gerald	3300	3630
50	149	10500 Eleni	2100	2310
51	150	10000 Peter	3000	3300
52	151	9500 David	2375	2612,5
53	152	9000 Peter	2250	2475
54	153	8000 Christopher	1600	1760

- Mostrar el contenido de la tabla de regiones con los encabezados de las columnas en español y el contenido de las columnas en mayúsculas.

Use la función UPPER para mostrar todo el contenido en mayúsculas

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Conexiones' pane shows the 'HR' schema with tables like COUNTRIES, DEPARTMENTS, EMPLOYEES, etc. The 'Hoja de Trabajo' pane contains the following SQL query:

```
END AS Proyeccion,  
CASE WHEN commission_pct IS NULL  
THEN 0  
ELSE  
(commission_pct * 0.10 + commission_pct) * Salary  
END AS Proyeccion10  
From employees;  
  
SELECT Region_id as ID_Región,  
UPPER(region_name) as Nombre_Región  
from regions;
```

The 'Resultado de la Consulta' pane shows the results of the query:

ID_REGION	NOMBRE_REGION
1	EUROPE
2	AMERICAS
3	ASIA
4	MIDDLE EAST AND AFRICA

- Mostrar los empleados del departamento 1

The screenshot shows the Oracle SQL Developer interface. The 'Hoja de Trabajo' pane contains the following SQL query:

```
ELSE  
(commission_pct * 0.10 + commission_pct) * Salary  
END AS Proyeccion10  
From employees;  
  
SELECT Region_id as ID_Región,  
UPPER(region_name) as Nombre_Región  
from regions;  
  
SELECT *  
FROM Employees  
where department_id = 10
```

The 'Resultado de la Consulta' pane shows the results of the query:

YEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	200	Jennifer	Whalen	JWHALEN	515.123.4444	17/09/03	AD_ASST	4400	(null)	101

8. Mostrar los empelados que no tienen jefe.

The screenshot shows the Oracle SQL Developer interface with the following content:

Hoja de Trabajo / Generador de Consultas:

```
SELECT Region_id as ID_Region,  
UPPER(region_name) as Nombre_Region  
from regions;  
  
SELECT *  
FROM Employees  
where department_id = 10;  
  
SELECT *  
FROM Employees  
where manager_id is null;
```

Resultado de la Consulta:

Todas las Filas Recuperadas: 1 en 0,003 segundos

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	Steven	King	SKING	515.123.4567	17/06/03	AD_PRES	24000	(null)	(null)	9

Historial SQL:

Historial SQL | Filtrar

9. Mostrar los empleados que no se les paga comisión

Oracle SQL Developer : HR

Archivo Editar Ver Navegar Ejecutar Origen Equipos Herramientas Ventana Ayuda

Página de bienvenida x HR

Hoja de Trabajo Generador de Consultas

```
SELECT *  
FROM Employees  
where department_id = 10;  
  
SELECT *  
FROM Employees  
where manager_id is null;  
  
SELECT *  
FROM Employees  
where commission_pct is null;
```

Salida de Script x Resultado de la Consulta x

Todas las Filas Recuperadas: 72 en 0,006 segundos

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPART
1	100	Steven	King	SKING	515.123.4567	17/06/03	AD_PRES	24000	(null)	(null)	
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21/09/05	AD_VP	17000	(null)	100	
3	102	Lex	De Haan	LDEHAAN	515.123.4569	13/01/01	AD_VP	17000	(null)	100	
4	103	Alexander	Hunold	AHUNOLD	590.423.4567	03/01/06	IT_PROG	9000	(null)	102	
5	104	Bruce	Ernst	BERNST	590.423.4568	21/05/07	IT_PROG	6000	(null)	103	
6	105	David	Austin	DAUSTIN	590.423.4569	25/06/05	IT_PROG	4800	(null)	103	
7	106	Valli	Pataballa	VPATABAL	590.423.4560	05/02/06	IT_PROG	4800	(null)	103	
8	107	Diana	Lorentz	DLORENTZ	590.423.5567	07/02/07	IT_PROG	4200	(null)	103	
9	108	Nancy	Greenberg	NGREENBE	515.124.4569	17/08/02	FI_MGR	12008	(null)	101	
10	109	Daniel	Faviet	DFAVIET	515.124.4169	16/08/02	FI_ACCOUNT	9000	(null)	108	
11	110	John	Chen	JCHEN	515.124.4269	28/09/05	FI_ACCOUNT	8200	(null)	108	
12	111	Ismael	Sciarra	ISCIARRA	515.124.4369	30/09/05	FI_ACCOUNT	7700	(null)	108	
13	112	Jose Manuel	Urman	JMURMAN	515.124.4469	07/03/06	FI_ACCOUNT	7800	(null)	108	
14	113	Luis	Popp	LPOPP	515.124.4567	07/12/07	FI_ACCOUNT	6900	(null)	108	
15	114	Den	Raphaely	DRAPHEAL	515.127.4561	07/12/02	PU_MAN	11000	(null)	100	
16	115	Alexander	Khoo	AKHOO	515.127.4562	18/05/03	PU_CLERK	3100	(null)	114	
17	116	Shelli	Baida	SBAIDA	515.127.4563	24/12/05	PU_CLERK	2900	(null)	114	
18	117	Stacy	Tobias	STOBIAS	515.127.4564	24/07/05	PU_CLERK	2800	(null)	114	

Historial SQL

10. Presentar los departamentos cuyo código es menor de 500.

The screenshot shows the Oracle SQL Developer interface with the 'HR' schema selected. The 'Conexiones' pane on the left displays the database structure, including tables like COUNTRIES, DEPARTMENTS, EMPLOYEES, and JOBS. The 'Hoja de Trabajo' pane shows a SQL query in the 'Generador de Consultas' tab. The query is as follows:

```
SELECT *  
FROM Employees  
where manager_id is null;  
  
SELECT *  
FROM Employees  
where commission_pct is null;  
  
SELECT *  
FROM departments  
where department_id < 500;
```

The 'Resultado de la Consulta' pane displays the results of the third query, showing 18 rows of department data. The status bar indicates 'Todas las Filas Recuperadas: 27 en 0,003 segundos'.

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10 Administration	200	1700
2	20 Marketing	201	1800
3	30 Purchasing	114	1700
4	40 Human Resources	203	2400
5	50 Shipping	121	1500
6	60 IT	103	1400
7	70 Public Relations	204	2700
8	80 Sales	145	2500
9	90 Executive	100	1700
10	100 Finance	108	1700
11	110 Accounting	205	1700
12	120 Treasury	(null)	1700
13	130 Corporate Tax	(null)	1700
14	140 Control And Credit	(null)	1700
15	150 Shareholder Services	(null)	1700
16	160 Benefits	(null)	1700
17	170 Manufacturing	(null)	1700
18	180 Construction	(null)	1700

11. Mostrar el nombre del departamento y la tercera letra del nombre.

Aquí utilice la función substr que te permite extraer una parte específica de una cadena de texto.

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Conexiones' pane displays a tree of database objects under the 'HR' schema, including tables like COUNTRIES, DEPARTMENTS, EMPLOYEES, and JOBS. The main window is titled 'Hoja de Trabajo' and contains a SQL query in the 'Generador de Consultas' pane. The query is as follows:

```
FROM Employees
where commission_pct is null;

SELECT *
FROM departments
where department_id < 500;

SELECT department_name,
substr(department_name,3,1) as Tercera_Letra
FROM departments
```

Below the query, the 'Resultado de la Consulta' pane shows the results of the query. It displays a table with two columns: 'DEPARTMENT_NAME' and 'TERCERA_LETRA'. The results are as follows:

DEPARTMENT_NAME	TERCERA_LETRA
1 Administration	m
2 Marketing	r
3 Purchasing	r
4 Human Resources	m
5 Shipping	i
6 IT	(null)
7 Public Relations	b
8 Sales	l
9 Executive	e
10 Finance	n
11 Accounting	c
12 Treasury	e
13 Corporate Tax	r
14 Control And Credit	n
15 Shareholder Services	a
16 Benefits	n
17 Manufacturing	n
18 Construction	n

12. Mostrar el nombre del departamento y la longitud del mismo.

Ahora, use la función Length para contar los caracteres

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Conexiones' pane displays a tree view of the database schema, including tables like COUNTRIES, DEPARTMENTS, EMPLOYEES, and JOBS. The 'Hoja de Trabajo' pane shows a SQL query being executed. The query is as follows:

```
where department_id < 500;

SELECT department_name,
substr(department_name,3,1) as Tercera_Letra
FROM departments

SELECT department_name,
length(department_name) as Longitud
FROM departments
```

The 'Resultado de la Consulta' pane displays the results of the query in a table format. The table has two columns: DEPARTMENT_NAME and LONGITUD. The results are as follows:

DEPARTMENT_NAME	LONGITUD
1 Administration	14
2 Marketing	9
3 Purchasing	10
4 Human Resources	15
5 Shipping	8
6 IT	2
7 Public Relations	16
8 Sales	5
9 Executive	9
10 Finance	7
11 Accounting	10
12 Treasury	8
13 Corporate Tax	13
14 Control And Credit	18
15 Shareholder Services	20
16 Benefits	8
17 Manufacturing	13
18 Construction	12

13. Presente nuevamente el resultado anterior pero ordenado por la longitud.

Con la función Order By es posible ordenar los elementos tanto de forma ascendente como descendente.

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema with tables like COUNTRIES, DEPARTMENTS, EMPLOYEES, and REGIONS. The main editor shows a SQL query that filters departments by ID and then lists them by length. The results pane at the bottom shows the output of the query, ordered by the length of the department names.

```
where department_id < 500;

SELECT department_name,
substr(department_name,3,1) as Tercera_Letra
FROM departments;

SELECT department_name,
length(department_name) as Longitud
FROM departments
ORDER BY Longitud;
```

DEPARTMENT_NAME	LONGITUD
1 IT	2
2 NOC	3
3 Sales	5
4 Payroll	7
5 Finance	7
6 Benefits	8
7 Treasury	8
8 Shipping	8
9 Executive	9
10 Marketing	9
11 Accounting	10
12 Purchasing	10
13 Recruiting	10
14 IT Support	10
15 Operations	10
16 IT Helpdesk	11
17 Contracting	11
18 Construction	12

14. Mostrar el código del departamento y el nombre, así como el contenido del nombre hasta la posición que corresponda a la línea desplegada.

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Conexiones' pane displays a tree view of the database schema, including tables like COUNTRIES, DEPARTMENTS, EMPLOYEES, and JOBS. The 'DEPARTMENTS' table is selected. The main window shows a SQL script in the 'Hoja de Trabajo' tab. The script consists of two queries. The first query is a simple 'FROM departments;' statement. The second query is a SELECT statement that retrieves department information and a substring of the department name. The results of the second query are displayed in the 'Resultado de la Consulta' pane at the bottom.

```
FROM departments;
```

```
SELECT department_name,  
length(department_name) as Longitud  
FROM departments  
ORDER BY Longitud;
```

```
SELECT department_id AS Codigo_Departamento,  
department_name AS Nombre_Departamento,  
SUBSTR(department_name, 1, ROWNUM) AS Nombre_Correspondiente  
FROM departments;
```

	CODIGO_DEPARTAMENTO	NOMBRE_DEPARTAMENTO	NOMBRE_CORRESPONDIENTE
1	10	Administration	A
2	20	Marketing	Ma
3	30	Purchasing	Pur
4	40	Human Resources	Huma
5	50	Shipping	Shipp
6	60	IT	IT
7	70	Public Relations	Public
8	80	Sales	Sales
9	90	Executive	Executive
10	100	Finance	Finance
11	110	Accounting	Accounting
12	120	Treasury	Treasury
13	130	Corporate Tax	Corporate Tax
14	140	Control And Credit	Control And Cr
15	150	Shareholder Services	Shareholder Ser
16	160	Benefits	Benefits
17	170	Manufacturing	Manufacturing
18	180	Construction	Construction

15. Se necesita generar un archivo de datos del contenido de la tabla de empleados. Este archivo es de tamaño fijo, por lo que debe de rellenar los campos numéricos con ceros a la derecha y los de texto con espacios a la derecha. La longitud de los campos en el archivo es la misma que tienen en la tabla.

The screenshot shows the Oracle SQL Developer interface with a query window titled 'Generador de Consultas'. The query is designed to format employee data into a fixed-width format using the `RPAD` function. The query is as follows:

```
select * from employees

SELECT RPAD(TO_CHAR(Employee_ID), LENGTH(TO_CHAR(Employee_ID)), '0') AS Employee_ID_Formato,
       RPAD(TO_CHAR(Salary), LENGTH(TO_CHAR(Salary)), '0') AS Salary_Formato,
       RPAD(TO_CHAR(Commission_PCT), LENGTH(TO_CHAR(Commission_PCT)), '0') AS Commission_Formato,
       RPAD(TO_CHAR(Manager_ID), LENGTH(TO_CHAR(Manager_ID)), '0') AS Manager_ID_Formato,
       RPAD(TO_CHAR(Department_ID), LENGTH(TO_CHAR(Department_ID)), '0') AS Department_ID_Formato,
       RPAD(First_Name, LENGTH(First_Name), ' ') AS First_Name_Formato,
       RPAD>Last_Name, LENGTH>Last_Name, ' ') AS Last_Name_Formato,
       RPAD(email, LENGTH(email), ' ') AS Email_Formato,
       RPAD(PHONE_NUMBER, LENGTH(PHONE_NUMBER), ' ') AS Phone_Number_Formato,
       RPAD(JOB_ID, LENGTH(JOB_ID), ' ') AS Job_ID_Formato
FROM employees;

describe employees
```

Below the query window, the 'Resultado de la Consulta' (Query Result) tab shows the output of the query. It displays a table with 5 columns: `EMPLOYEE_ID_FORMATO`, `SALARY_FORMATO`, `COMMISSION_FORMATO`, `MANAGER_ID_FORMATO`, and `DEPARTMENT_ID_FORMATO`. The results show 10 rows of data, with the first row being 49 148 11000 ,3 100 80.

EMPLOYEE_ID_FORMATO	SALARY_FORMATO	COMMISSION_FORMATO	MANAGER_ID_FORMATO	DEPARTMENT_ID_FORMATO
49 148	11000	,3	100	80
50 149	10500	,2	100	80
51 150	10000	,3	145	80
52 151	9500	,25	145	80
53 152	9000	,25	145	80
54 153	8000	,2	145	80
55 154	7500	,2	145	80
56 155	7000	,15	145	80
57 156	10000	,35	146	80
58 157	9500	,35	146	80
59 158	9000	,35	146	80
60 159	8000	,3	146	80
61 160	7500	,3	146	80

16. Mostrar los diferentes títulos de trabajos(tabla jobs), de las siguientes manera: todo en mayúscula, todo en minúscula, y las primeras 5 posiciones del nombre.

Todo en mayúscula

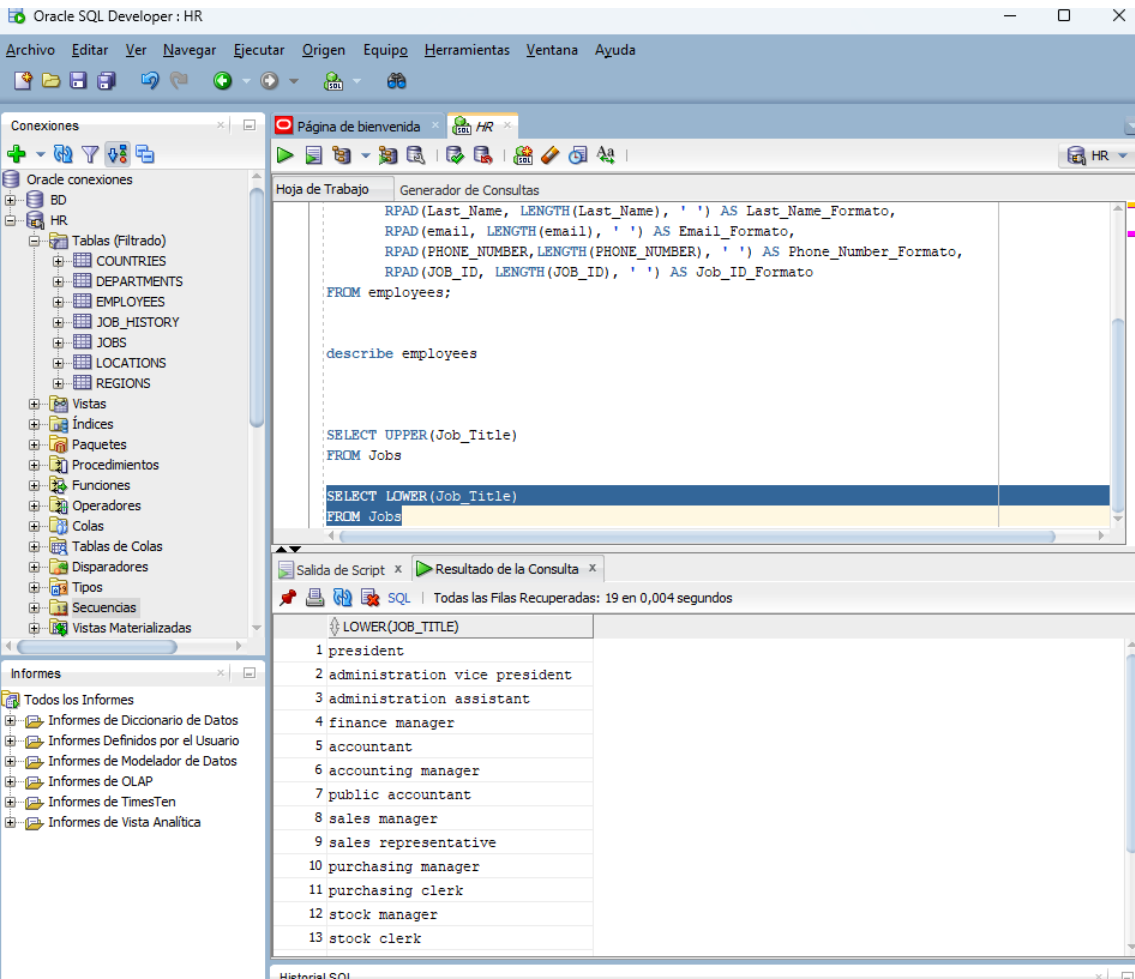
The screenshot shows the Oracle SQL Developer interface. The left pane displays the 'Conexiones' (Connections) tree with the 'HR' schema selected. The main editor shows a SQL script with the following content:

```
RPAD(TO_CHAR(Manager_ID), LENGTH(TO_CHAR(Manager_ID)), '0') AS Manager_ID_Formato,  
RPAD(TO_CHAR(Department_ID), LENGTH(TO_CHAR(Department_ID)), '0') AS Department_ID_Formato,  
RPAD(First_Name, LENGTH(First_Name), ' ') AS First_Name_Formato,  
RPAD>Last_Name, LENGTH>Last_Name), ' ') AS Last_Name_Formato,  
RPAD(email, LENGTH(email), ' ') AS Email_Formato,  
RPAD(PHONE_NUMBER, LENGTH(PHONE_NUMBER), ' ') AS Phone_Number_Formato,  
RPAD(JOB_ID, LENGTH(JOB_ID), ' ') AS Job_ID_Formato  
FROM employees;  
  
describe employees  
  
SELECT UPPER(Job Title)  
FROM Jobs
```

The bottom pane shows the 'Resultado de la Consulta' (Query Result) window, displaying the results of the query. The results are as follows:

UPPER(JOB_TITLE)
1 PRESIDENT
2 ADMINISTRATION VICE PRESIDENT
3 ADMINISTRATION ASSISTANT
4 FINANCE MANAGER
5 ACCOUNTANT
6 ACCOUNTING MANAGER
7 PUBLIC ACCOUNTANT
8 SALES MANAGER
9 SALES REPRESENTATIVE
10 PURCHASING MANAGER
11 PURCHASING CLERK
12 STOCK MANAGER
13 STOCK CLERK

Todo en minúscula



5 primeros registros

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema for the HR schema, including tables like COUNTRIES, DEPARTMENTS, EMPLOYEES, JOB_HISTORY, JOBS, LOCATIONS, and REGIONS. The main editor window contains the following SQL query:

```
FROM employees;  
  
describe employees  
  
SELECT UPPER(Job_Title)  
FROM Jobs  
  
SELECT LOWER(Job_Title)  
FROM Jobs  
  
SELECT (Job_Title)  
FROM Jobs  
where rownum <= 5
```

The bottom pane shows the results of the query, displaying the first 5 records of the JOB_TITLE column:

JOB_TITLE
1 President
2 Administration Vice President
3 Administration Assistant
4 Finance Manager
5 Accountant

17. Presentar la información de los nombres y apellidos de los empleados. Las vocales del nombre se deben sustituir por * y las de los apellidos por @.

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Conexiones' pane shows a tree view of the database schema, including tables like COUNTRIES, DEPARTMENTS, EMPLOYEES, and JOBS. The 'EMPLOYEES' table is selected. The main window displays a SQL query in the 'Hoja de Trabajo' (Worksheet) tab. The query is as follows:

```

SELECT UPPER(Job_Title)
FROM Jobs

SELECT LOWER(Job_Title)
FROM Jobs

SELECT (Job_Title)
FROM Jobs
where rownum <= 5

select First_Name, Last_Name,
TRANSLATE(First_Name, 'aeiou', '*') as Nombre_Reemplazado,
TRANSLATE(Last_Name, 'aeiou', '@') as apellido_Reemplazado
from employees
  
```

Below the query, the 'Resultado de la Consulta' (Query Result) pane shows the output of the query. It indicates that 50 rows were retrieved in 0.003 seconds. The results are displayed in a table with the following columns: FIRST_NAME, LAST_NAME, NOMBRE_REEMPLAZADO, and APELLIDO_REEMPLAZADO. The first 13 rows are shown in the screenshot.

	FIRST_NAME	LAST_NAME	NOMBRE_REEMPLAZADO	APELLIDO_REEMPLAZADO
1	Ellen	Abel	Elln	Ab1
2	Sundar	Ande	Snd*r	And
3	Mozhe	Atkinson	Mzh	Atknsn
4	David	Austin	D*vd	Astn
5	Hermann	Baer	Hrm*nn	B@r
6	Shelli	Baida	Sh11	B@d@
7	Amit	Banda	Amt	B@end@
8	Elizabeth	Bates	Elz*bth	B@ts
9	Sarah	Bell	S*r*h	B11
10	David	Bernstein	D*vd	Brnstn
11	Laura	Bissot	L*r*	Bsst
12	Harrison	Bloom	H*rrsn	Blm
13	Alexis	Bull	Alye	B11

18. Presentar la cantidad de días, y la cantidad de meses que tienen laborando los empleados de la empresa.

The screenshot shows the Oracle SQL Developer interface. The 'Conexiones' pane on the left shows the 'HR' schema with tables like COUNTRIES, DEPARTMENTS, EMPLOYEES, JOB_HISTORY, JOBS, LOCATIONS, and REGIONS. The 'Hoja de Trabajo' pane contains the following SQL query:

```
SELECT * FROM EMPLOYEES
```

The 'Resultado de la Consulta' pane shows the results of the query, displaying 50 rows. The columns are EMPLOYEE_ID, FIRST_NAME, HIRE_DATE, NUMERO_MESES, and NUMERO_DIAS. The data is as follows:

EMPLOYEE_ID	FIRST_NAME	HIRE_DATE	NUMERO_MESES	NUMERO_DIAS
1	100 Steven	17/06/03	256	7804
2	101 Neena	21/09/05	229	6977
3	102 Lex	13/01/01	285	8689
4	103 Alexander	03/01/06	225	6873
5	104 Bruce	21/05/07	209	6370
6	105 David	25/06/05	232	7065
7	106 Valli	05/02/06	224	6840
8	107 Diana	07/02/07	212	6473
9	108 Nancy	17/08/02	266	8108
10	109 Daniel	16/08/02	266	8109

19. Presentar el empleado y su salario. El salario debe ser rellenado de # a la derecha y colocarle el símbolo de pesos. En total se deben presentar 20 posiciones en todos los casos.

The screenshot shows the Oracle SQL Developer interface. The left pane displays the 'Conexiones' (Connections) tree with 'HR' selected. The main editor shows a SQL query. The bottom pane shows the 'Resultado de la Consulta' (Query Result) window with 10 rows of data.

Query:

```
SELECT Employee_ID, First_Name, Hire_Date,
TRUNC(MONTHS_BETWEEN(SYSDATE, HIRE_DATE)) AS Numero_meses,
TRUNC(SYSDATE - HIRE_DATE) AS Numero_dias
FROM employees

SELECT first_name || ' ' || last_name AS Nombre_Empleado,
RPAD('$' || TO_CHAR(salary, '999,999.99'), 20, '#') AS Salario_Formato
FROM employees;
```

Query Result:

	NOMBRE_EMPLEADO	SALARIO_FORMATO
1	Steven King	\$ 24,000.00#####
2	Neena Kochhar	\$ 17,000.00#####
3	Lex De Haan	\$ 17,000.00#####
4	Alexander Hunold	\$ 9,000.00#####
5	Bruce Ernst	\$ 6,000.00#####
6	David Austin	\$ 4,800.00#####
7	Valli Pataballa	\$ 4,800.00#####
8	Diana Lorentz	\$ 4,200.00#####
9	Nancy Greenberg	\$ 12,008.00#####
10	Daniel Faviot	\$ 9,000.00#####

20. Mostrar los empleados, y el nombre del departamento en el que laboran.

The screenshot shows the Oracle SQL Developer interface. On the left, the 'Conexiones' pane displays a tree view of the database schema, including tables like DEPARTMENT_ID, DEPARTMENT_NAME, MANAGER_ID, LOCATION_ID, EMPLOYEES, EMPLOYEE_ID, FIRST_NAME, LAST_NAME, EMAIL, PHONE_NUMBER, HIRE_DATE, JOB_ID, SALARY, COMMISSION_PCT, MANAGER_ID, DEPARTMENT_ID, JOB_HISTORY, and JOBS. The 'Informes' pane at the bottom left lists various report types. The main workspace is divided into two panes: 'Hoja de Trabajo' (Worksheet) and 'Generador de Consultas' (Query Generator). The 'Generador de Consultas' pane contains the following SQL query:

```
select employees.First_Name, Employees.Department_ID, Department_NAME
from employees
INNER JOIN departments ON employees.department_ID = departments.department_ID
```

Below the query, the 'Resultado de la Consulta' (Query Result) pane displays the results of the query. It shows a table with three columns: FIRST_NAME, DEPARTMENT_ID, and DEPARTMENT_NAME. The table contains 13 rows of data, representing employees and their respective departments.

FIRST_NAME	DEPARTMENT_ID	DEPARTMENT_NAME
1 Jennifer	10	Administration
2 Pat	20	Marketing
3 Michael	20	Marketing
4 Sigal	30	Purchasing
5 Karen	30	Purchasing
6 Shelli	30	Purchasing
7 Den	30	Purchasing
8 Alexander	30	Purchasing
9 Guy	30	Purchasing
10 Susan	40	Human Resources
11 Kevin	50	Shipping
12 Jean	50	Shipping
13 Adam	50	Shipping

21. Repetir el ejercicio anterior, pero en este caso los departamentos sin empleados deben mostrarse.

The screenshot shows the Oracle SQL Developer interface. The 'Conexiones' pane on the left displays the database schema with tables like COUNTRIES, DEPARTMENTS, EMPLOYEES, and JOB_HISTORY. The 'Hoja de Trabajo' pane contains two SQL queries. The first query uses an INNER JOIN, and the second query uses a LEFT JOIN. The 'Resultado de la Consulta' pane shows the results of the second query, displaying 107 rows of employee data, including names and department names. The last row shows a null department name for Kimberly.

```
select employees.First_Name, Employees.Department_ID, Department_NAME
from employees
INNER JOIN departments ON employees.department_ID = departments.department_ID

select employees.First_Name, Departments.Department_NAME
from employees
LEFT JOIN departments on employees.department_ID = departments.Department_ID
```

FIRST_NAME	DEPARTMENT_NAME
93 Jonathon	Sales
94 Jack	Sales
95 Charles	Sales
96 Steven	Executive
97 Neena	Executive
98 Lex	Executive
99 Nancy	Finance
100 Daniel	Finance
101 John	Finance
102 Ismael	Finance
103 Jose Manuel	Finance
104 Luis	Finance
105 Shelley	Accounting
106 William	Accounting
107 Kimberly	(null)

22. Modificar este ejercicio para agregar la información de las localidades. Se quiere ver el nombre de las mismas.

Herramientas Ventana Ayuda

Página de bienvenida x HR x REGIONS x

Hoja de Trabajo Generador de Consultas

```
select employees.First_Name, Employees.Department_ID, Department_NAME
from employees
INNER JOIN departments ON employees.department_ID = departments.department_ID

select employees.First_Name, Departments.Department_NAME
from employees
LEFT JOIN departments on employees.department_ID = departments.Department_ID

SELECT employees.first_name AS Nombre_Empleado,
       departments.department_name AS Nombre_Departamento,
       departments.location_id AS Location_ID,
       locations.state_province AS Provincia
FROM employees
LEFT JOIN departments ON employees.department_id = departments.department_id
LEFT JOIN locations ON departments.location_id = locations.location_id;
```

Resultado de la Consulta x

SQL | Se han recuperado 50 filas en 0,008 segundos

	NOMBRE_EMPLEADO	NOMBRE_DEPARTAMENTO	LOCATION_ID	PROVINCIA
1	Alexander	IT	1400	Texas
2	Bruce	IT	1400	Texas
3	David	IT	1400	Texas
4	Valli	IT	1400	Texas
5	Diana	IT	1400	Texas
6	Matthew	Shipping	1500	California
7	Adam	Shipping	1500	California
8	Payam	Shipping	1500	California
9	Shanta	Shipping	1500	California
10	Kevin	Shipping	1500	California
11	Julia	Shipping	1500	California
12	Irene	Shipping	1500	California
13	James	Shipping	1500	California
14	Steven	Shipping	1500	California
15	Laura	Shipping	1500	California

23. Mostrar el empleado, su fecha de contratación y su fecha de evaluación. Se supone que esta se realiza seis meses después de su contratación.

Herramientas Ventana Ayuda

Página de bienvenida REGIONES

Hoja de Trabajo Generador de Consultas

```

SELECT employees.first_name AS Nombre_Empleado,
       departments.department_name AS Nombre_Departamento,
       departments.location_id AS Location_ID,
       locations.state_province AS Provincia
FROM employees
LEFT JOIN departments ON employees.department_id = departments.department_id
LEFT JOIN locations ON departments.location_id = locations.location_id;

SELECT first_name || ' ' || last_name AS Nombre_Empleado,
       hire_date AS Fecha_Contratacion,
       ADD_MONTHS(hire_date, 6) AS Fecha_Evaluacion
FROM employees;

```

Resultado de la Consulta

Se han recuperado 50 filas en 0,006 segundos

	NOMBRE_EMPLEADO	FECHA_CONTRATACION	FECHA_EVALUACION
1	Steven King	17/06/03	17/12/03
2	Neena Kochhar	21/09/05	21/03/06
3	Lex De Haan	13/01/01	13/07/01
4	Alexander Russell	03/01/06	03/07/06
5	Bruce Ernst	21/05/07	21/11/07
6	David Austin	25/06/05	25/12/05
7	Valli Pataballa	05/02/06	05/08/06
8	Diana Lorentz	07/02/07	07/08/07
9	Nancy Greenberg	17/08/02	17/02/03
10	Daniel Faviet	16/08/02	16/02/03
11	John Chen	28/09/05	28/03/06
12	Ismael Sciarra	30/09/05	31/03/06
13	Jose Manuel Urman	07/03/06	07/09/06
14	Luis Popp	07/12/07	07/06/08
15	Den Raphaely	07/12/02	07/06/03

Historial SQL

24. Determinar el promedio de salario por departamento, así como el máximo salario pagado y el mínimo.

Herramientas Ventana Ayuda

Página de bienvenida REGIONES

Hoja de Trabajo Generador de Consultas

```

FROM employees
LEFT JOIN departments ON employees.department_id = departments.department_id
LEFT JOIN locations ON departments.location_id = locations.location_id;

SELECT first_name || ' ' || last_name AS Nombre_Empleado,
       hire_date AS Fecha_Contratacion,
       ADD_MONTHS(hire_date, 6) AS Fecha_Evaluacion
FROM employees;

SELECT department_id AS Departamento,
       TRUNC(AVG(salary)) AS Promedio_Salario,
       MAX(salary) AS Maximo_Salario,
       MIN(salary) AS Minimo_Salario
FROM employees
GROUP BY department_id;

```

Resultado de la Consulta

Todas las Filas Recuperadas: 12 en 0,004 segundos

	DEPARTAMENTO	PROMEDIO_SALARIO	MAXIMO_SALARIO	MINIMO_SALARIO
1	90	19333	24000	17000
2	60	5760	9000	4200
3	100	8601	12008	6900
4	30	4150	11000	2500
5	50	3475	8200	2100
6	80	8955	14000	6100
7	(null)	7000	7000	7000
8	10	4400	4400	4400
9	20	9500	13000	6000
10	40	6500	6500	6500
11	70	10000	10000	10000
12	110	10154	12008	8300

25. Indicar el código y nombre o nombres de los empleados que ganan mas dinero por departamento.

```

e.department_id
FROM EMPLOYEES e
JOIN
(SELECT department_id, MAX(salary) AS max_salary
FROM EMPLOYEES
GROUP BY department_id) max_salaries
ON e.department_id = max_salaries.department_id
AND e.salary = max_salaries.max_salary

```

Query Result x

SQL | All Rows Fetched: 11 in 0.126 seconds

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	SALARY	DEPARTMENT_ID
1	100	Steven	King	24000	90
2	103	Alexander	Hunold	9000	60
3	108	Nancy	Greenberg	12008	100
4	114	Den	Raphaely	11000	30
5	121	Adam	Fripp	8200	50
6	145	John	Russell	14000	80
7	200	Jennifer	Whalen	4400	10
8	201	Michael	Hartstein	13000	20
9	203	Susan	Mavris	6500	40
10	204	Hermann	Beer	10000	70
11	205	Shelley	Higgins	12008	110

25. Presentar el nombre del departamento, su código y el total de salario ganado por los empleados de este.

```

SELECT
d.department_name AS nombre_departamento,
d.department_id AS codigo_departamento,
SUM(e.salary) AS total_salario
FROM DEPARTMENTS d
JOIN EMPLOYEES e ON d.department_id = e.department_id
GROUP BY d.department_name, d.department_id

```

Query Result x

SQL | All Rows Fetched: 11 in 0.031 seconds

	NOMBRE_DEPARTAMENTO	CODIGO_DEPARTAMENTO	TOTAL_SALARIO
1	Administration	10	4400
2	Marketing	20	19000
3	Purchasing	30	24900
4	Human Resources	40	6500
5	Shipping	50	156400
6	IT	60	28800
7	Public Relations	70	10000
8	Sales	80	304500
9	Executive	90	58000
10	Finance	100	51608
11	Accounting	110	20308

26. Determinar el promedio de salario por localidad.

```
SELECT locations.location_id AS Localidad,
       locations.city AS Ciudad,
       trunc(AVG(employees.salary)) AS Promedio_Salario
FROM employees
INNER JOIN departments ON employees.department_id = departments.department_id
INNER JOIN locations ON departments.location_id = locations.location_id
GROUP BY locations.location_id, locations.city;
```

Resultado de la Consulta x

Todas las Filas Recuperadas: 7 en 0,017 segundos

	LOCALIDAD	CIUDAD	PROMEDIO_SALARIO
1	1700	Seattle	8845
2	1400	Southlake	5760
3	1500	South San Francisco	3475
4	2500	Oxford	8955
5	1800	Toronto	9500
6	2400	London	6500
7	2700	Munich	10000

27. Determinar el empleado que mas veces ha cambiado de trabajo. Auxiliarse de la tabla Job History.

```
SELECT employee_id AS Codigo_Empleado,
       COUNT(*) AS Cambios_Trabajo
FROM job_history
GROUP BY employee_id
ORDER BY Cambios_Trabajo DESC
FETCH FIRST 1 ROWS ONLY;
```

Resultado de la Consulta x

Todas las Filas Recuperadas: 1 en 0,008 s

	CODIGO_EMPLEADO	CAMBIOS_TRABAJO
1	101	2

28. Determinar la cantidad de empleados que existen por el tipo de labor que desempeñan.

```
SELECT job_id AS Tipo_Labor,  
       COUNT(*) AS Cantidad_Empleados  
FROM employees  
GROUP BY job_id  
ORDER BY Cantidad_Empleados DESC;
```

Resultado de la Consulta x

Todas las Filas Recuperadas: 19 en 0,005

TIPO_LABOR	CANTIDAD_EMPLEADOS
1 SA_REP	30
2 SH_CLERK	20
3 ST_CLERK	20
4 PU_CLERK	5
5 FI_ACCOUNT	5
6 SA_MAN	5
7 IT_PROG	5
8 ST_MAN	5
9 AD_VP	2
10 PU_MAN	1
11 PR_REP	1

29. Indicar el nombre del empleado cuya longitud de nombre es la mayor.

```
SELECT first_name || ' ' || last_name AS Nombre_Empleado,  
       LENGTH(first_name || ' ' || last_name) AS Longitud_Nombre  
FROM employees  
ORDER BY Longitud_Nombre DESC  
FETCH FIRST 1 ROWS ONLY;
```

Resultado de la Consulta x

Todas las Filas Recuperadas: 1 en 0,006 segundos

NOMBRE_EMPLEADO	LONGITUD_NOMBRE
1 Nanette Cambrault	17

30. Crear una salida en formato de matriz que muestre la información de los empleados por departamento y el rango de salario que ganan. Tenemos tres rangos de 1 a 3000, de 3000 a 4999 y de 5000 o mas.

```

SELECT department_name AS Departamento,
       COUNT(CASE WHEN salary BETWEEN 1 AND 3000 THEN 1 END) AS "Rango 1 a 3000",
       COUNT(CASE WHEN salary BETWEEN 3001 AND 4999 THEN 1 END) AS "Rango 3000 a 4999",
       COUNT(CASE WHEN salary >= 5000 THEN 1 END) AS "De 5000 o más"
FROM employees
JOIN departments ON employees.department_id = departments.department_id
GROUP BY department_name
ORDER BY department_name;

```

Resultado de la Consulta x

Todas las Filas Recuperadas: 11 en 0,006 segundos

DEPARTAMENTO	Rango 1 a 3000	Rango 3000 a 4999	De 5000 o más
1 Accounting	0	0	2
2 Administration	0	1	0
3 Executive	0	0	3
4 Finance	0	0	6
5 Human Resources	0	0	1
6 IT	0	3	2
7 Marketing	0	0	2
8 Public Relations	0	0	1
9 Purchasing	4	1	1
10 Sales	0	0	34
11 Shipping	22	18	5

31. Mostrar el nombre del departamento y la cantidad de empleados por departamento, así como el máximo y mínimo salario.

```

SELECT departments.department_name AS Departamento,
       COUNT(employees.employee_id) AS Cantidad_Empleados,
       MAX(employees.salary) AS Maximo_Salario,
       MIN(employees.salary) AS Minimo_Salario
FROM employees
JOIN departments ON employees.department_id = departments.department_id
GROUP BY departments.department_name
ORDER BY departments.department_name;

```

Resultado de la Consulta x

Todas las Filas Recuperadas: 11 en 0,006 segundos

DEPARTAMENTO	CANTIDAD_EMPLEADOS	MAXIMO_SALARIO	MINIMO_SALARIO
1 Accounting	2	12008	8300
2 Administration	1	4400	4400
3 Executive	3	24000	17000
4 Finance	6	12008	6900
5 Human Resources	1	6500	6500
6 IT	5	9000	4200
7 Marketing	2	13000	6000
8 Public Relations	1	10000	10000
9 Purchasing	6	11000	2500
10 Sales	34	14000	6100
11 Shipping	45	8200	2100

32. Encontrar la cantidad de empleados por región

```
SELECT regions.region_name AS Region,
       COUNT(employees.employee_id) AS Cantidad_Empleados
FROM employees
JOIN departments ON employees.department_id = departments.department_id
JOIN locations ON departments.location_id = locations.location_id
JOIN countries ON locations.country_id = countries.country_id
JOIN regions ON countries.region_id = regions.region_id
GROUP BY regions.region_name
ORDER BY regions.region_name;
```

Resultado de la Consulta x

Todas las Filas Recuperadas: 2 en 0,052 segundos

REGION	CANTIDAD_EMPLEADOS
1 Americas	70
2 Europe	36

33. Mostrar la fecha del día, presentando el nombre del día y del mes en vez de los números.

```
SELECT TO_CHAR(SYSDATE, 'Day, DD "de" Month YYYY') AS Fecha_Actual
FROM dual;
```

Resultado de la Consulta x

Todas las Filas Recuperadas: 1 en 0,003 segundos

FECHA_ACTUAL
1 Martes , 29 de Octubre 2024

34. Mostrar los empleados que ganan mas que el promedio de salario del departamento en que laboran.

```
SELECT e.employee_id AS Codigo_Empleado,
       e.first_name || ' ' || e.last_name AS Nombre_Empleado,
       e.department_id AS Departamento,
       e.salary AS Salario
FROM employees e
JOIN (
  SELECT department_id,
        AVG(salary) AS Promedio_Salario
  FROM employees
  GROUP BY department_id
) d ON e.department_id = d.department_id
WHERE e.salary > d.Promedio_Salario
ORDER BY e.department_id;
```

Resultado de la Consulta x

Todas las Filas Recuperadas: 38 en 0,012 segundos

CODIGO_EMPLEADO	NOMBRE_EMPLEADO	DEPARTAMENTO	SALARIO
1	201 Michael Hartstein	20	13000
2	114 Den Raphaely	30	11000
3	141 Tenna Rajs	50	3500
4	137 Renske Ladwig	50	3600
5	189 Jennifer Dilly	50	3600
6	188 Kelly Chung	50	3800
7	193 Britney Everett	50	3900
8	192 Sarah Bell	50	4000
9	185 Alexis Bull	50	4100
10	184 Nandita Sarchand	50	4200
11	124 Kevin Mourgos	50	5800
12	123 Shanta Vollman	50	6500
13	122 Payam Kaufling	50	7900
14	120 Matthew Weiss	50	8000
15	121 Adam Frippe	50	8200

35. Indicar los empleados que ganan menos de 1/3 del salario que gana king

```

SELECT
  employee_id,
  first_name || ' ' || last_name AS nombre,
  salary
FROM EMPLOYEES
WHERE
  salary < (SELECT MAX(salary) / 3
            FROM EMPLOYEES
            WHERE last_name = 'King')

```

Query Result

SQL | Fetched 50 rows in 0.011 seconds

EMPLOYEE_ID	NOMBRE	SALARY
1	104 Bruce Ernst	6000
2	105 David Austin	4800
3	106 Valli Pataballa	4800
4	107 Diana Lorentz	4200
5	111 Ismael Sciarra	7700
6	112 Jose Manuel Urman	7800
7	113 Luis Popp	6900
8	115 Alexander Khoo	3100
9	116 Shelli Baida	2900
10	117 Sigal Tobias	2800
11	118 Guy Himuro	2600
12	119 Karen Colmenares	2500
13	120 Payan Kaufling	7900
14	123 Shanta Vollman	6500
15	124 Kevin Mourgos	5600
16	125 Julia Nayer	3200
17	126 Irene Mikilineni	2700
18	127 James Landry	2400
19	128 Steven Markle	2200
20	129 Laura Bissot	3300
21	130 Moshe Atkinson	2800
22	131 James Marlow	2500
23	132 TJ Olson	2100
24	133 Thomas Mather	2000

36. Mostrar el nombre del empleado, su código, salario y el nombre del departamento en que labora utilizando subconsultas.

```

SELECT
  first_name || ' ' || last_name AS Nombre_Empleado,
  employee_id AS Codigo_Empleado,
  salary AS Salario,
  (SELECT department_name
   FROM departments
   WHERE departments.department_id = employees.department_id) AS Nombre_Departamento
FROM employees;

```

Resultado de la Consulta

SQL | Se han recuperado 50 filas en 0,004 segundos

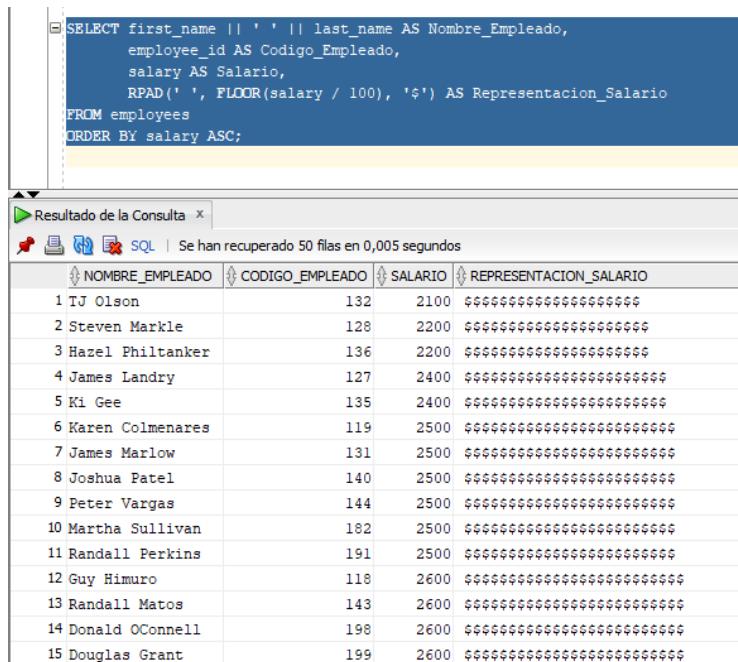
NOMBRE_EMPLEADO	CODIGO_EMPLEADO	SALARIO	NOMBRE_DEPARTAMENTO
1 Steven King	100	24000	Executive
2 Neena Kochhar	101	17000	Executive
3 Lex De Haan	102	17000	Executive
4 Alexander Hunold	103	9000	IT
5 Bruce Ernst	104	6000	IT
6 David Austin	105	4800	IT
7 Valli Pataballa	106	4800	IT
8 Diana Lorentz	107	4200	IT
9 Nancy Greenberg	108	12008	Finance
10 Daniel Faviat	109	9000	Finance
11 John Chen	110	8200	Finance
12 Ismael Sciarra	111	7700	Finance
13 Jose Manuel Urman	112	7800	Finance
14 Luis Popp	113	6900	Finance
15 Den Raphaely	114	11000	Purchasing

37. Indicar cuales son los principales n clientes utilizando subconsultas.

No hay ninguna tabla con esa información.

39. Indicar cuales son los principales n departamentos por el total de salario que se ganan en este utilizando subconsultas.

40. Desplegar el nombre salario, nombre, apellido y codigo de los empleados. Adicionalmente mostrar la cantidad de dinero que gana representado con un grupo de signos de pesos a la derecha del salario, cada 100 en salario representan un signo de pesos.

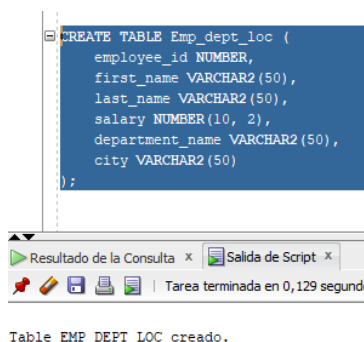


The screenshot shows a SQL query in a text editor and its execution results in a table. The query selects employee details and formats the salary with peso symbols. The results table has 15 rows, ordered by salary in ascending order.

```
SELECT first_name || ' ' || last_name AS Nombre_Empleado,
       employee_id AS Codigo_Empleado,
       salary AS Salario,
       RPAD(' ', FLOOR(salary / 100), '$') AS Representacion_Salario
FROM employees
ORDER BY salary ASC;
```

NOMBRE_EMPLEADO	CODIGO_EMPLEADO	SALARIO	REPRESENTACION_SALARIO
1 TJ Olson	132	2100	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
2 Steven Markle	128	2200	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
3 Hazel Philtanker	136	2200	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
4 James Landry	127	2400	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
5 Ki Gee	135	2400	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
6 Karen Colmenares	119	2500	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
7 James Marlow	131	2500	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
8 Joshua Patel	140	2500	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
9 Peter Vargas	144	2500	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
10 Martha Sullivan	182	2500	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
11 Randall Perkins	191	2500	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
12 Guy Himuro	118	2600	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
13 Randall Matos	143	2600	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
14 Donald OConnell	198	2600	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$
15 Douglas Grant	199	2600	\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$

41. Crear la tabla Emp_dept_loc con la información correspondiente a el código, nombre, apellido, salario, nombre del deptamento que trabaja, y nombre de la ciudad, en que labora el empleado.



The screenshot shows a SQL script to create a table named Emp_dept_loc. Below the script, the execution result is displayed, confirming the table's creation.

```
CREATE TABLE Emp_dept_loc (
  employee_id NUMBER,
  first_name VARCHAR2(50),
  last_name VARCHAR2(50),
  salary NUMBER(10, 2),
  department_name VARCHAR2(50),
  city VARCHAR2(50)
);
```

Resultado de la Consulta x Salida de Script x
Tarea terminada en 0,129 segundos

Table EMP_DEPT_LOC creado.

42. Agregue una columna a la tabla anterior llamada Full_name que sea varchar2 de 60.
Llenar esta columna con la información resultante de concatenar el nombre un espacio y el apellido.

```
ALTER TABLE Emp_dept_loc
ADD Full_name VARCHAR2(60);

UPDATE Emp_dept_loc
SET Full_name = first_name || ' ' || last_name;

SELECT employee_id, first_name, last_name, Full_name
FROM Emp_dept_loc;
```

Salida de Script x	Resultado de la Consulta x
SQL	Todas las Filas Recuperadas: 0 en 0,003 segundos
EMPLOYEE...	FIRST_NA...
LAST_NAME	FULL_NAME

43. Agregar una columna llamada hire_date a la tabla, Emp_dept_loc.

```
ALTER TABLE Emp_dept_loc
ADD hire_date DATE;
```

44. Actualizar la columna hire_date, con la información de la columna hire_date de la tabla de empleados (employees).

	CODIGO	NOMBRE	APELLIDO	SALARIO	DEPARTAMENTO	CIUDAD	FULL_NAME	HIRE_DATE
1	100	Steven	King	24000	Executive	Seattle	Steven King	17-JUN-03
2	101	Neena	Kochhar	17000	Executive	Seattle	Neena Kochhar	21-SEP-05
3	102	Lex	De Haan	17000	Executive	Seattle	Lex De Haan	13-JAN-01
4	103	Alexander	Hunold	9000	IT	Southlake	Alexander Hunold	03-JAN-06
5	104	Bruce	Ernst	6000	IT	Southlake	Bruce Ernst	21-MAY-07
6	105	David	Austin	4800	IT	Southlake	David Austin	25-JUN-05
7	106	Valli	Pataballa	4800	IT	Southlake	Valli Pataballa	05-FEB-06
8	107	Diana	Lorents	4200	IT	Southlake	Diana Lorents	07-FEB-07
9	108	Nancy	Greenberg	12008	Finance	Seattle	Nancy Greenberg	17-AUG-02
10	109	Daniel	Faviet	9000	Finance	Seattle	Daniel Faviet	16-AUG-02
11	110	John	Chen	8200	Finance	Seattle	John Chen	28-SEP-05
12	111	Ismael	Sciarra	7700	Finance	Seattle	Ismael Sciarra	30-SEP-05
13	112	Jose	Manuel Urman	7800	Finance	Seattle	Jose Manuel Urman	07-MAR-06
14	113	Luis	Popp	6900	Finance	Seattle	Luis Popp	07-DEC-07
15	114	Den	Raphaely	11000	Purchasing	Seattle	Den Raphaely	07-DEC-02
16	115	Alexander	Khoo	3100	Purchasing	Seattle	Alexander Khoo	18-MAY-03

45. Agregue un nuevo empleado a la tabla emp_dept_loc. Coloque la información relacionada a usted para completar los datos, para el codigo de empelado, utilice 9999 y la fecha de contratación con la fecha de sistema.

```
INSERT INTO Emp_dept_loc (employee_id, first_name, last_name, salary, department_name, city, hire_date, full_name)
VALUES (9999, 'Isaac', 'Fermin', 5000, 'IT', 'SantoDomingo', SYSDATE, 'Isaac Fermin');
```

46. Actualice el empleado creado y colóquele un aumento de un 20%

```
UPDATE Emp_dept_loc
SET salary = salary * 1.20
WHERE employee_id = 9999;
```

46. Elimine el empleado creado en la pregunta 45.

```
DELETE FROM Emp_dept_loc
WHERE employee_id = 9999;
```

47. Cree una nueva tabla llamada emp_dept_loc2, con la información de la tabla de emp_dept_loc, pero solamente para los empleado que ganan menos del promedio de salario de la empresa.

```
CREATE TABLE emp_dept_loc2 AS
SELECT *
FROM Emp_dept_loc
WHERE salary < (SELECT AVG(salary) FROM Emp_dept_loc);
```

Table EMP_DEPT_LOC2 creado.

49. Actualice la tabla emp_dept_loc2 y coloque el apellido y nombre todo en minúscula.

```
UPDATE emp_dept_loc2
SET first_name = LOWER(first_name),
    last_name = LOWER(last_name);
```

	CODIGO	NOMBRE	APELLIDO	SALARIO	DEPARTAMENTO	CIUDAD	FULL_NAME	HIRE_DATE
1	104	bruce	ernst	6000	IT	Southlake	Bruce Ernst	21-MAY-07
2	105	david	austin	4800	IT	Southlake	David Austin	25-JUN-05
3	106	valli	patabella	4800	IT	Southlake	Valli Patabella	05-FEB-06
4	107	diana	lorents	4200	IT	Southlake	Diana Lorents	07-FEB-07
5	115	alexander	khoo	3100	Purchasing	Seattle	Alexander Khoo	18-MAY-03
6	116	shell	baida	2900	Purchasing	Seattle	Shell Baida	24-DEC-05
7	117	sigal	tobias	2800	Purchasing	Seattle	Sigal Tobias	24-JUL-05
8	118	guy	himuro	2600	Purchasing	Seattle	Guy Himuro	15-NOV-06
9	119	karen	colmenares	2500	Purchasing	Seattle	Karen Colmenares	10-AUG-07
10	124	kevin	mourgos	5800	Shipping	South San Francisco	Kevin Mourgos	16-NOV-07
11	125	julia	nayer	3200	Shipping	South San Francisco	Julia Nayer	16-JUL-05
12	126	irene	mikkilineni	2700	Shipping	South San Francisco	Irene Mikkilineni	28-SEP-06
13	127	james	landry	2400	Shipping	South San Francisco	James Landry	14-JAN-07
14	128	steven	markle	2200	Shipping	South San Francisco	Steven Markle	08-MAR-08

50. Actualice la información de emp_dept_loc2 usando como fuente emp_dept_loc. Para esto utilice el comando merge. Solamente actualice la información del nombre y agregue los empelados que no existen en emp_dept_loc. Verifique que fuesen exitosas las actualizaciones.

```

MERGE INTO emp_dept_loc2 target
USING Emp_dept_loc source
ON (target.employee_id = source.employee_id)
WHEN MATCHED THEN
    UPDATE SET target.first_name = source.first_name,
               target.last_name = source.last_name
WHEN NOT MATCHED THEN
    INSERT (employee_id, first_name, last_name, salary, department_name, city, hire_date, full_name)
VALUES (source.employee_id, source.first_name, source.last_name, source.salary, source.department_name, source.city,

```

EMPLOYEE_ID	CODIGO	NOMBRE	APELLIDO	SALARIO	DEPARTAMENTO	CIUDAD	FULL NAME	HIRE DATE
83	122	Vayam	Kautiling	7900	Shipping	South San Francisco	Vayam Kautiling	01-MAY-03
84	148	Gerald	Cambrault	11000	Sales	Oxford	Gerald Cambrault	15-OCT-07
85	170	Taylor	Fox	9600	Sales	Oxford	Taylor Fox	24-JAN-06
86	149	Eleni	Zlotkey	10500	Sales	Oxford	Eleni Zlotkey	29-JAN-08
87	112	Jose M...	Urman	7800	Finance	Seattle	Jose Manuel Urman	07-MAR-06
88	146	Karen	Partners	13500	Sales	Oxford	Karen Partners	05-JAN-05
89	152	Peter	Hall	9000	Sales	Oxford	Peter Hall	20-AUG-05
90	171	William	Smith	7400	Sales	Oxford	William Smith	23-FEB-07
91	111	Ismael	Sciarra	7700	Finance	Seattle	Ismael Sciarra	30-SEP-05
92	153	Christ...	Olsen	8000	Sales	Oxford	Christopher Olsen	30-MAR-06
93	164	Mattea	Marvins	7200	Sales	Oxford	Mattea Marvins	24-JAN-08
94	201	Michael	Hartstein	13000	Marketing	Toronto	Michael Hartstein	17-FEB-04
95	114	Den	Raphaely	11000	Purchasing	Seattle	Den Raphaely	07-DEC-02
96	154	Nanette	Cambrault	7500	Sales	Oxford	Nanette Cambrault	09-DEC-06
97	145	David	Lee	6800	Sales	Oxford	David Lee	23-FEB-08
98	176	Jonathon	Taylor	8600	Sales	Oxford	Jonathon Taylor	24-MAR-06
99	205	Shelley	Higgins	12008	Accounting	Seattle	Shelley Higgins	07-JUN-02
100	100	Steven	King	24000	Executive	Seattle	Steven King	17-JUN-03
101	123	Shanta	Vollman	4500	Shipping	South San Francisco	Shanta Vollman	10-OCT-05
102	175	Alyssa	Hutton	8800	Sales	Oxford	Alyssa Hutton	19-MAR-05
103	102	Lex	De Haan	17000	Executive	Seattle	Lex De Haan	13-JAN-01
104	147	Alberto	Errazuriz	12000	Sales	Oxford	Alberto Errazuriz	10-MAR-05
105	150	Peter	Tucker	10000	Sales	Oxford	Peter Tucker	30-JAN-05
106	160	Louise	Doran	7500	Sales	Oxford	Louise Doran	15-DEC-05

51. Presentar la siguiente salida, haciendo una consulta sobre la tabla emp_dept_loc. El empleado NOMBRE_COMPLETO labora en la empresa desde el FECHA_CONTRATACION y gana un sueldo de SALARIO. En donde Nombre_completo, FECHA_CONTRATACION y SALARIO son columnas de la tabla.

```

SELECT 'El empleado ' || full_name || ' labora en la empresa desde el ' ||
       TO_CHAR(hire_date, 'DD-MM-YYYY') || ' y gana un sueldo de ' ||
       salary AS mensaje
FROM emp_dept_loc;

```

DESCRIPCION
1 El empleado Steven King labora en la empresa desde el 17-Jun-2003 y gana un sueldo de 24000
2 El empleado Neena Kochhar labora en la empresa desde el 21-Sep-2005 y gana un sueldo de 17000
3 El empleado Lex De Haan labora en la empresa desde el 13-Jan-2001 y gana un sueldo de 17000
4 El empleado Alexander Runold labora en la empresa desde el 03-Jan-2006 y gana un sueldo de 9000
5 El empleado Bruce Ernst labora en la empresa desde el 21-May-2007 y gana un sueldo de 6000
6 El empleado David Austin labora en la empresa desde el 25-Jun-2005 y gana un sueldo de 4800
7 El empleado Valli Pataballa labora en la empresa desde el 05-Feb-2006 y gana un sueldo de 4800
8 El empleado Diana Lorentz labora en la empresa desde el 07-Feb-2007 y gana un sueldo de 4200
9 El empleado Nancy Greenberg labora en la empresa desde el 17-Aug-2002 y gana un sueldo de 12008
10 El empleado Daniel Faviat labora en la empresa desde el 16-Aug-2002 y gana un sueldo de 9000
11 El empleado John Chen labora en la empresa desde el 28-Sep-2005 y gana un sueldo de 8200
12 El empleado Ismael Sciarra labora en la empresa desde el 30-Sep-2005 y gana un sueldo de 7700
13 El empleado Jose Manuel Urman labora en la empresa desde el 07-Mar-2006 y gana un sueldo de 7800
14 El empleado Luis Popp labora en la empresa desde el 07-Dec-2007 y gana un sueldo de 6900
15 El empleado Den Raphaely labora en la empresa desde el 07-Dec-2002 y gana un sueldo de 11000

52. Eliminar la tabla emp_dept_loc2 de la base de datos.

```
DROP TABLE emp_dept_loc2;
```

53. Crear una vista llamada emp_dept_loc2, con la información correspondiente a el código, nombre, apellido, salario, nombre del departamento que trabaja, y nombre de la ciudad, en que labora el empleado.

```
CREATE VIEW emp_dept_loc2 AS
SELECT
    e.employee_id AS Codigo_Empleado,
    e.first_name AS Nombre,
    e.last_name AS Apellido,
    e.salary AS Salario,
    d.department_name AS Departamento,
    l.city AS Ciudad
FROM employees e
JOIN departments d ON e.department_id = d.department_id
JOIN locations l ON d.location_id = l.location_id;
```

View EMP_DEPT_LOC2 creado.

54. Cree una nueva sesión a la base de datos con el mismo usuario con que esta conectado. Ahora en esta nueva sesión borre la información de la tabla emp_dept_loc, pero no haga commit. En la sesión vieja haga un count(*) a la tabla emp_dept_loc. Cual fue el resultado de la consulta y porque. Ahora haga commit en la sesión nueva y en la vieja realice nuevamente el count. Que paso aquí y por que.

Antes del COMMIT: La sesión antigua ve el estado de la tabla como estaba antes del DELETE.

```
DELETE FROM emp_dept_loc;
```

Después del COMMIT: La sesión antigua ve el cambio, mostrando que la tabla está vacía (0 filas). Esto se debe al nivel de aislamiento de las transacciones en Oracle, que garantiza la consistencia de lectura en cada sesión.

```
SELECT COUNT(*) FROM emp_dept_loc;
```

COUNT(*)
0

55. Realice una consulta en la cual se muestre la estructura jerárquica de la empresa. Recuerde que el campo manager_id, muestra el código de empleado de mi jefe. La información debe mostrarse indentada atendiendo a la jerarquía del empleado, mientras jerárquicamente mas abajo se encuentre, mayor indentacion tiene.

```
SELECT LPAD(' ', LEVEL * 3, ' ') || first_name || ' ' || last_name AS Nombre_Empleado,
       employee_id AS Codigo_Empleado,
       manager_id AS Codigo_Jefe,
       LEVEL AS Nivel_Jerarquia
FROM employees
START WITH manager_id IS NULL
CONNECT BY PRIOR employee_id = manager_id
ORDER SIBLINGS BY first_name;
```

	◊ NOMBRE_EMPLEADO	◊ CODIGO_EMPLEADO	◊ CODIGO_JEFE	◊ NIVEL_JERARQUIA	
1	Steven King	100	(null)	1	
2	Adam Fripp	121	100	2	
3	Alexis Bull	185	121	3	
4	Anthony Cabrio	187	121	3	
5	James Marlow	131	121	3	
6	Julia Dellinger	186	121	3	
7	Laura Bissot	129	121	3	
8	Mozhe Atkinson	130	121	3	
9	Nandita Sarchand	184	121	3	
10	TJ Olson	132	121	3	
11	Alberto Errazuriz	147	100	2	
12	Amit Banda	167	147	3	
13	Clara Vishney	162	147	3	
14	Danielle Greene	163	147	3	
15	David Lee	165	147	3	