

Instituto FOC
Módulo de desarrollo de aplicaciones web

BRUNO MARENCO CERQUEIRA

Tarea Individual 4: Sentencia SELECT de consulta de datos

Enero/2018

Índice

Índice.....	2
Planteamiento	3
Se pide realizar las siguientes consultas:	3
Primera consulta.	3
Segunda consulta.....	4
Tercera consulta.....	4
Cuarta consulta.	5
Quinta consulta.	6
Sexta consulta.....	6
Séptima consulta.	7
Octava consulta.....	8
Novena consulta.	8
Décima consulta.....	9
Undécima consulta.	10
Duodécima consulta.....	11
Decimotercera consulta.....	11
Decimacuarta consulta.....	12

Planteamiento

El objetivo principal de esta tarea es la realización de consultas simples sobre un esquema de la base de datos, es decir, se realizarán consultas en la que sólo se mostrarán datos referentes a una única tabla y la utilización de funciones resumen.

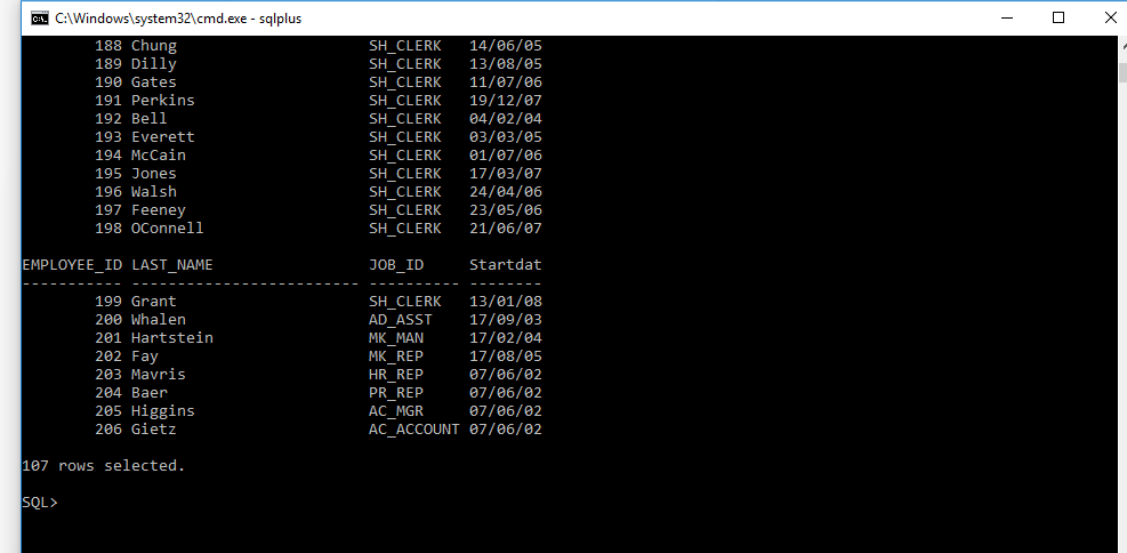
Para la realización de esta tarea, se deberá usar el esquema “HR” con el que se viene trabajando a lo largo del curso. Para acceder al esquema “HR” de Oracle como administrador (con las credenciales usadas en la máquina virtual de la asignatura) se deben usar las siguientes sentencias:

Se pide realizar las siguientes consultas:

Primera consulta.

- El departamento de recursos humanos desea que una consulta muestre el apellido, ID del cargo (job_id), fecha de contratación e ID de empleado de cada empleado (el atributo ID del empleado debe aparecer en primer lugar). Proporcione un alias STARTDATE para la columna HIRE_DATE.

SELECT employee_id, last_name, job_id, hire_date AS “Startdate” FROM employees;



```

C:\Windows\system32\cmd.exe - sqlplus

188 Chung                SH_CLERK    14/06/05
189 Dilly                SH_CLERK    13/08/05
190 Gates                SH_CLERK    11/07/06
191 Perkins              SH_CLERK    19/12/07
192 Bell                 SH_CLERK    04/02/04
193 Everett              SH_CLERK    03/03/05
194 McCain               SH_CLERK    01/07/06
195 Jones                SH_CLERK    17/03/07
196 Walsh                SH_CLERK    24/04/06
197 Feeney               SH_CLERK    23/05/06
198 OConnell             SH_CLERK    21/06/07

EMPLOYEE_ID LAST_NAME      JOB_ID      Startdat
-----
199 Grant      SH_CLERK    13/01/08
200 Whalen     AD_ASST     17/09/03
201 Hartstein  MK_MAN      17/02/04
202 Fay        MK_REP      17/08/05
203 Mavris     HR_REP      07/06/02
204 Baer       PR_REP      07/06/02
205 Higgins    AC_MGR      07/06/02
206 Gietz      AC_ACCOUNT  07/06/02

107 rows selected.

SQL>

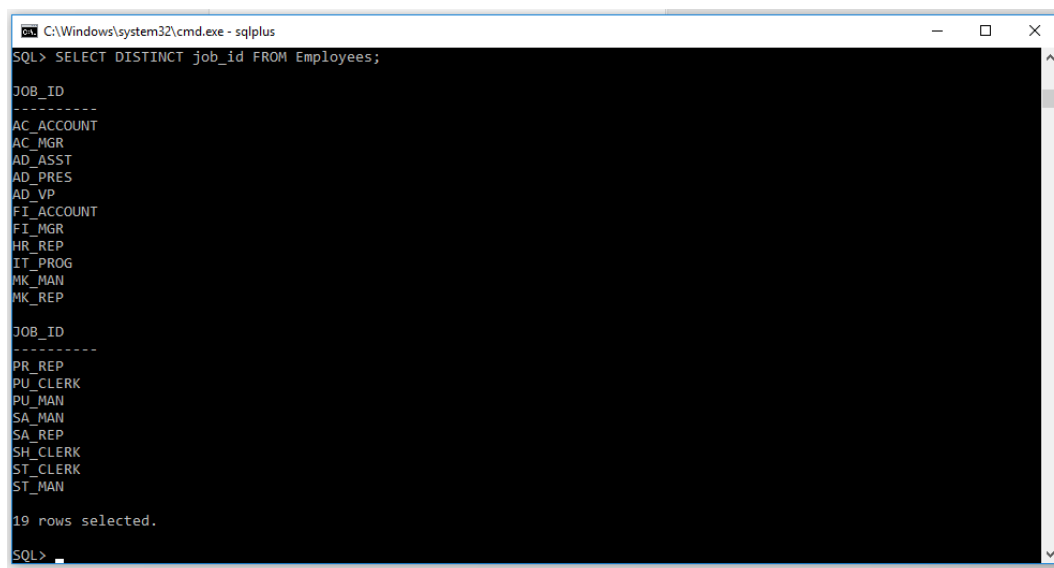
```

Se realiza un *Select* para realizar la consulta, indicando primero los atributos de la consulta por orden (con el ID del empleado en primer lugar), aplicando un alias a la columna *HIRE_DATE* a través del comando *AS* y por último, se determina la tabla donde se realiza la consulta con el comando *FROM*.

Segunda consulta.

- El departamento de recursos humanos desea una consulta para mostrar todos los ID de cargos (*job_id*) únicos de la tabla *Employees*.

SELECT DISTINCT job_id FROM Employees;



```

C:\Windows\system32\cmd.exe - sqlplus
SQL> SELECT DISTINCT job_id FROM Employees;

JOB_ID
-----
AC_ACCOUNT
AC_MGR
AD_ASST
AD PRES
AD_VP
FI_ACCOUNT
FI_MGR
HR REP
IT_PROG
MK MAN
MK REP
PR REP
PU_CLERK
PU MAN
SA MAN
SA REP
SH_CLERK
ST_CLERK
ST MAN

19 rows selected.

SQL>

```

En este ejemplo usamos el comando *DISTINCT* antes de atributo para mostrar los valores únicos, eliminando duplicados de la lista.

Tercera consulta.

- El departamento de recursos humanos ha solicitado un informe de todos los empleados y el id de su cargo (*job_id*). Muestre el apellido concatenado con el ID del cargo (separado por una coma y un espacio) y asigne el nombre 'Employee and Title'.

SELECT last_name || ', ' || job_id AS "Employee and Title" FROM Employees;

```

C:\Windows\system32\cmd.exe - sqlplus
Seo, ST_CLERK
Employee and Title
-----
Sewall, SA_REP
Smith, SA_REP
Smith, SA_REP
Stiles, ST_CLERK
Sullivan, SH_CLERK
Sully, SA_REP
Taylor, SA_REP
Taylor, SH_CLERK
Tobias, PU_CLERK
Tucker, SA_REP
Tuvault, SA_REP
Employee and Title
-----
Urman, FI_ACCOUNT
Vargas, ST_CLERK
Vishney, SA_REP
Vollman, ST_MAN
Walsh, SH_CLERK
Weiss, ST_MAN
Whalen, AD_ASST
Zlotkey, SA_MAN
107 rows selected.
SQL>

```

En este caso utilizamos || para realizar la concatenación de los datos en la consulta, concatenando el apellido y el cargo por con una coma en medio. Se renombra la lista con el comando AS para 'Employee and Title'.

Cuarta consulta.

- Debido a problemas presupuestarios, el departamento de recursos humanos necesita un informe que muestre el apellido y el salario de los empleados que ganan más de 12000 €.

SELECT last_name, salary FROM Employees WHERE salary>12000;

```

C:\Windows\system32\cmd.exe - sqlplus
LAST_NAME          SALARY
-----
King                24000
Kochhar             17000
De Haan             17000
Greenberg           12008
Russell             14000
Partners            13500
Hartstein           13000
Higgins             12008
8 rows selected.

SQL> SELECT last_name, salary FROM Employees WHERE salary>12000;

LAST_NAME          SALARY
-----
King                24000
Kochhar             17000
De Haan             17000
Greenberg           12008
Russell             14000
Partners            13500
Hartstein           13000
Higgins             12008
8 rows selected.

SQL>

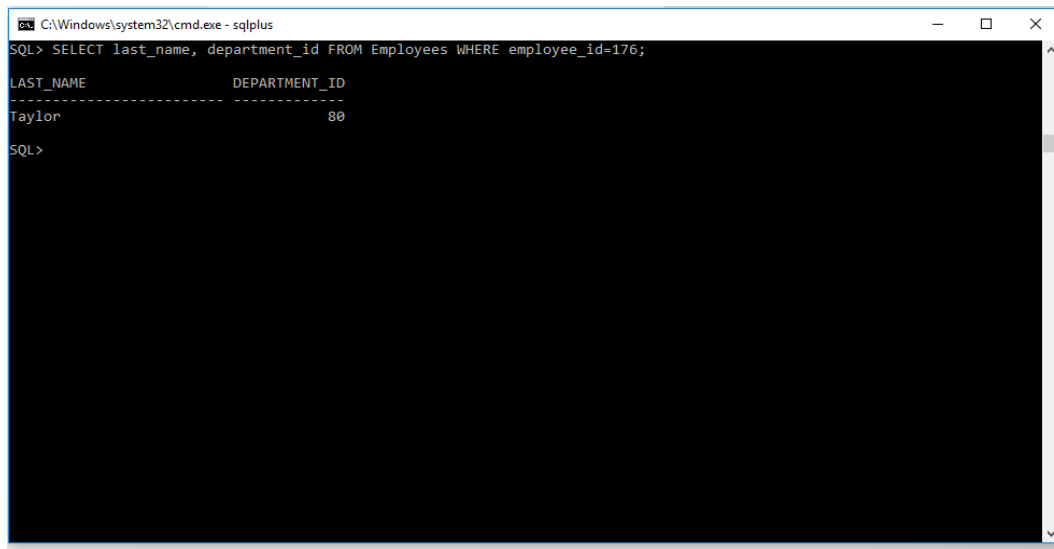
```

Utilizamos el comando WHERE para aplicar la condición solicitada, en este caso, que el salario sea mayor que 12.000,00.

Quinta consulta.

- Cree una consulta que muestre el apellido y el número del departamento, para el empleado cuyo id es el 176.

```
SELECT last_name, department_id FROM Employees WHERE employee_id=176;
```



```
C:\Windows\system32\cmd.exe - sqlplus
SQL> SELECT last_name, department_id FROM Employees WHERE employee_id=176;

LAST_NAME      DEPARTMENT_ID
-----
Taylor          80

SQL>
```

Utilizamos el comando WHERE para aplicar la condición solicitada, en este caso, que el empleado sea el que tenga el id 176.

Sexta consulta.

- El departamento de recursos humanos necesita encontrar los empleados con salarios tanto muy altos como muy bajos, para ello se debe mostrar el apellido y el salario de cualquier empleado cuyo salario NO esté entre 5000 y 12000 €.

```
SELECT last_name, salary FROM Employees WHERE salary NOT BETWEEN 5000 AND 12000;
```

```

C:\Windows\system32\cmd.exe - sqlplus
Geoni                2800
Sarchand             4200
Bull                 4100
Dellinger            3400
Cabrio               3000
Chung                3800
Dilly                3600

LAST_NAME            SALARY
-----
Gates                2900
Perkins              2500
Bell                 4000
Everett              3900
McCain               3200
Jones                2800
Walsh                3100
Feeney               3000
OConnell             2600
Grant                2600
Whalen               4400

LAST_NAME            SALARY
-----
Hartstein            13000
Higgins              12008

57 rows selected.

SQL>

```

Utilizamos el comando *WHERE* para aplicar la condición solicitada, en este caso se piden los datos de los empleados cuyos salarios no estén entre 5.000,00 y 12.000,00, por lo que utilizamos el comando *NOT* (para que se devuelvan los que no cumplan la condición a seguir) *BETWEEN* (que devuelve los valores de un rango) y separamos el rango por *AND*.

Séptima consulta.

- Cree un informe para mostrar el apellido, ID del cargo y fecha de contratación de los empleados cuyos apellidos sean Matos y Taylor. Ordene la consulta en orden ascendente por fecha de contratación.

SELECT last_name, job_id, hire_date FROM Employees WHERE last_name='Matos' OR last_name='Taylor' ORDER BY hire_date ASC;

```

C:\Windows\system32\cmd.exe - sqlplus

SQL> SELECT last_name, job_id, hire_date FROM Employees WHERE last_name='Matos' OR last_name='Taylor' ORDER BY hire_date
ASC;

LAST_NAME            JOB_ID    HIRE_DAT
-----
Taylor                SH_CLERK  24/01/06
Matos                 ST_CLERK  15/03/06
Taylor                SA_REP    24/03/06

SQL>

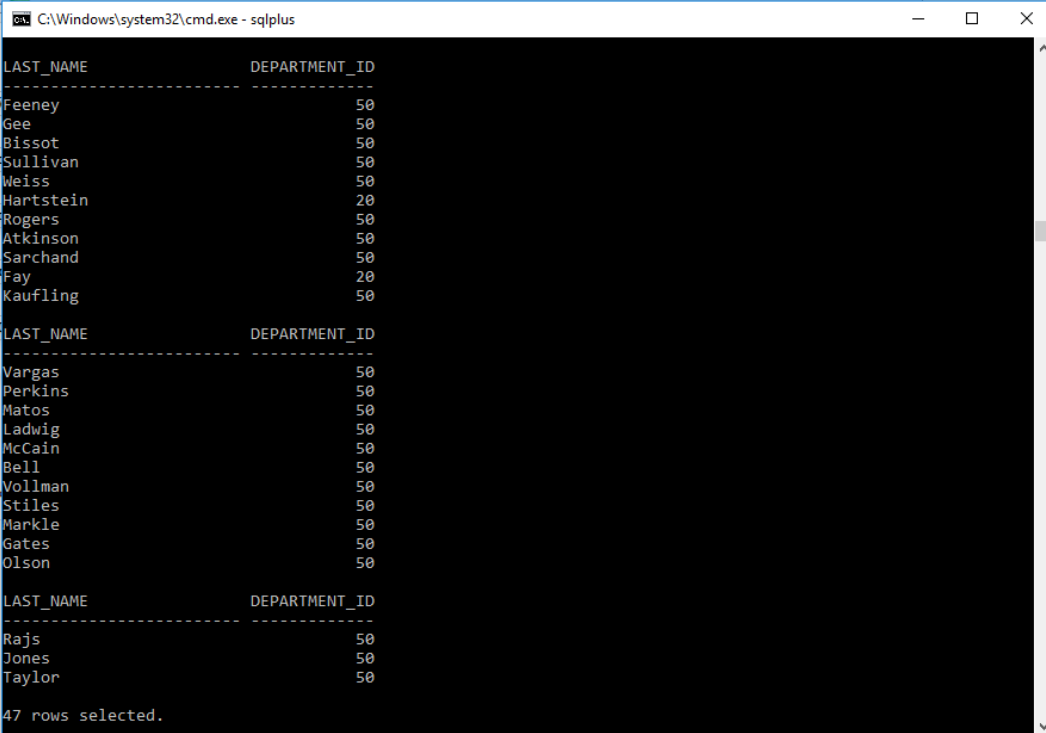
```

Utilizamos el comando *WHERE* para aplicar la condición solicitada, en este caso se piden los datos de los empleados cuyos apellidos sean Matos y Taylor y que esté ordenado de forma ascendente por la fecha de contratación, para lo que usamos el comando *ASC* en ese atributo.

Octava consulta.

- Muestre el apellido y el ID de departamento de todos los empleados de los departamentos 20 o 50 en orden alfabético ascendente por nombre.

```
SELECT last_name, department_id FROM Employees WHERE  
department_id='20' OR department_id='50' ORDER BY first_name ASC;
```



LAST_NAME	DEPARTMENT_ID
Feeney	50
Gee	50
Bissot	50
Sullivan	50
Weiss	50
Hartstein	20
Rogers	50
Atkinson	50
Sarchand	50
Fay	20
Kaufling	50
Vargas	50
Perkins	50
Matos	50
Ladwig	50
McCain	50
Bell	50
Vollman	50
Stiles	50
Markle	50
Gates	50
Olson	50
Rajes	50
Jones	50
Taylor	50

47 rows selected.

Utilizamos el comando *WHERE* para aplicar la condición solicitada, en este caso se piden los datos de los empleados cuyos departamentos sean el 20 o el 50, por lo que usamos el comando *OR* entre las dos condiciones (*department_id=20* y *department_id=50*) y que esté ordenado de forma ascendente por el nombre, para lo que usamos el comando *ASC* en ese atributo.

Novena consulta.

- Muestre el apellido y el salario de los empleados que ganan entre 5000 y 12000 € y que están en el departamento 20 o 50.

```
SELECT last_name, salary FROM Employees WHERE (salary BETWEEN 5000  
AND 12000) AND (department_id IN (20, 50));
```



```

C:\Windows\system32\cmd.exe - sqlplus
6 rows selected.
SQL> SELECT last_name, salary FROM Employees WHERE (salary BETWEEN 5000 AND 12000) AND (department_id IN (20, 50));

LAST_NAME          SALARY
-----
Weiss               8000
Fripp               8200
Kaufling            7900
Vollman             6500
Mourgos             5800
Fay                 6000

6 rows selected.
SQL>

```

Utilizamos el comando *WHERE* para aplicar la condición solicitada, en este caso se establecen dos condiciones, por lo que cada condición las colocamos entre paréntesis y las separamos por *AND* (dado que las dos condiciones se tienen que cumplir). La primera condición pide que el salario esté entre 5.000,00 y 12.000,00, para lo que utilizamos *BETWEEN*. La segunda condición establece que sean de cualquiera de dos departamentos el 20 y el 50, para lo que se usa *IN*.

Décima consulta.

- Muestre los apellidos de los empleados que tengan una “a” y una “e” en su apellido.

SELECT last_name FROM Employees WHERE last_name LIKE '%a%' AND last_name LIKE '%e%';

```

C:\Windows\system32\cmd.exe - sqlplus
SQL> SELECT last_name FROM Employees WHERE last_name LIKE '%a%' AND last_name LIKE '%e%';

LAST_NAME
-----
Baer
Bates
Colmenares
Davies
De Haan
Faviet
Fleaur
Gates
Hartstein
Markle
Nayer

LAST_NAME
-----
Partners
Patel
Philtanker
Raphaely
Sewall
Whalen

17 rows selected.
SQL>

```

Utilizamos el comando *WHERE* para aplicar la condición solicitada, en este caso se piden los datos de los empleados que tengan una “a” y una “e” en su apellido, para ello utilizamos el comando *LIKE* que compara los valores buscados con una expresión y si son iguales, devuelve el valor, por ello, se comparan los valores con ‘%a%’ y ‘%e%’, donde la expresión % antes y después, nos indica que puede existir cualquier carácter antes y después de “a” y “e”, encontrando las palabras que contengan las dos letras.

Undécima consulta.

- Muestre el apellido, cargo y salario de todos los empleados, que sean vendedores (SA_REP) u oficinistas (ST_CLERK) y cuyo salario no sea igual que 2500, 3000 o 7000 €.

SELECT last_name, job_id, salary FROM Employees WHERE (job_id='SA_REP' OR job_id='ST_CLERK') AND (salary NOT IN (2500, 3000, 7000));

```

C:\Windows\system32\cmd.exe - sqlplus
Bernstein      SA_REP      9500
Hall           SA_REP      9000
Olsen          SA_REP      8000
Cambrault      SA_REP      7500

LAST_NAME      JOB_ID      SALARY
-----
King           SA_REP      10000
Sully          SA_REP      9500
McEwen         SA_REP      9000
Smith          SA_REP      8000
Doran          SA_REP      7500
Vishney        SA_REP      10500
Greene         SA_REP      9500
Marvins        SA_REP      7200
Lee            SA_REP      6800
Aude           SA_REP      6400
Banda          SA_REP      6200

LAST_NAME      JOB_ID      SALARY
-----
Ozer           SA_REP      11500
Bloom          SA_REP      10000
Fox            SA_REP      9600
Smith          SA_REP      7400
Bates          SA_REP      7300
Kumar          SA_REP      6100
Abel           SA_REP      11000
Hutton         SA_REP      8800
Taylor         SA_REP      8600
Livingston     SA_REP      8400
Johnson       SA_REP      6200

44 rows selected.

SQL>

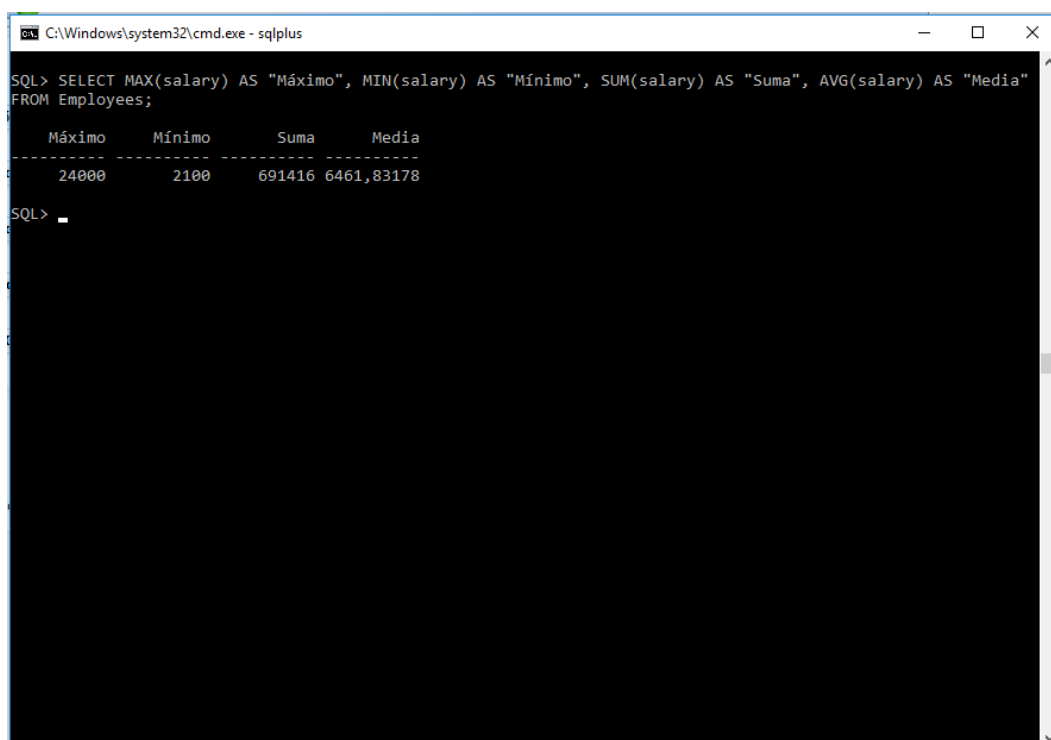
```

Utilizamos el comando *WHERE* para aplicar la condición solicitada, en este caso se establecen dos condiciones, por lo que agrupamos cada condición dentro de un paréntesis, y como las dos condiciones son exigidas, los separamos por un *AND*. La primera condición pide los datos de quien sea vendedor u oficinista, para lo cual separamos las dos comparaciones del cargo con *OR*, porque no son exigibles al mismo tiempo. La segunda condición pide los salarios que no sean 2.500,00, 3.000,00 y 7.000,00. Para ello utilizamos el *NOT* (para devolver el valor siempre que no se cumpla la condición a seguir) y el *IN* (que devuelve la consulta cuando es igual a cualquiera de los valores del grupo).

Duodécima consulta.

- Encontrar el valor más alto, el valor más bajo, la suma y la media del salario de todos los empleados. Etiquete las columnas como Máximo, Mínimo, Suma y Media, respectivamente.

SELECT MAX(salary) AS "Máximo", MIN(salary) AS "Mínimo", SUM(salary) AS "Suma", AVG(salary) AS "Media" FROM Employees;



```

C:\Windows\system32\cmd.exe - sqlplus
SQL> SELECT MAX(salary) AS "Máximo", MIN(salary) AS "Mínimo", SUM(salary) AS "Suma", AVG(salary) AS "Media"
FROM Employees;

  Máximo    Mínimo      Suma      Media
-----
  24000      2100    691416  6461,83178

SQL>
  
```

Utilizamos el comando *MAX* seguido del atributo para consultar el máximo valor de ese atributo, *MIN* seguido del atributo para consultar el menor valor, *SUM* seguido del atributo para consultar el valor de la suma de los valores de ese atributo y el *AVG* seguido del atributo para consultar la media. A cada valor se le aplica *AS* para editar el nombre de cada columna.

Decimotercera consulta.

- Determine el número de jefes (*manager_id*) que sean distintos. Etiquete la columna como "Number of Managers".

SELECT COUNT (DISTINCT *manager_id*) AS 'Number of Managers' FROM Employees;

```

C:\Windows\system32\cmd.exe - sqlplus
ORA-00923: FROM keyword not found where expected

SQL> SELECT COUNT (DISTINCT manager_id) AS "Number of Managers" FROM Employees;

Number of Managers
-----
                18

SQL>

```

Utilizamos *COUNT* para devolver el número de valores del atributo pedido, en este caso jefes (*manager_id*). Usando *DISTINCT* contamos valores únicos, sin duplicados. También usamos *AS* para editar el nombre de la tabla para "Number of Managers".

Decimacuarta consulta.

- Busque la diferencia entre los salarios más altos y más bajos. Etiquete la columna como *DIFFERENCE*.

SELECT MAX(salary) - MIN(salary) AS "DIFFERENCE" FROM Employees;

```

C:\Windows\system32\cmd.exe - sqlplus
21900

SQL> SELECT MAX(salary) - MIN(salary) AS "DIFFERENCE" FROM Employees;

DIFFERENCE
-----
        21900

SQL>

```

Usamos *MAX* y *MIN* en cada atributo para buscar los valores máximo y mínimo del mismo. Para tener como resultado la diferencia entre estos valores, hacemos uso de un operador, en este caso el meno "-" para que nos devuelva la diferencia entre el máximo y el mínimo de salario. También aplicamos *AS* para editar la columna con el nombre de "DIFFERENCE".