

PRACTICAS SQL (Usuario SCOTT)

1. Ver las tablas del esquema SCOTT

```
SELECT *  
FROM cat
```

2. Seleccionar todas las columnas de la tabla de empleados

```
SELECT *  
FROM emp;
```

3. Seleccionar el nombre, oficio, sueldo y comisión de los empleados (usar alias)

```
SELECT ename nombre, job oficio, sal salario, comm comision  
FROM emp;
```

4. Calcular cuantas semanas completas ha trabajado cada empleado

```
SELECT ename nombre, FLOOR((sysdate-hiredate)/7) semanas  
FROM emp;
```

O también:

```
SELECT ename nombre, TRUNC((sysdate-hiredate)/7) semanas  
FROM emp;
```

5. Calcular el sueldo neto de cada empleado, y la retención IRPF mensual, sabiendo que la retención es del 19% para el sueldo y de un 10% de la comisión, (redondear a 2 decimales)

```
SELECT ename nombre, sal salario, NVL(comm,0) comision,  
       ROUND ((sal*0.19)+(nvl(comm,0)*0.10),2) "Retención IRPF",  
       (sal+nvl(comm,0)- ROUND ((sal*0.19)+(nvl(comm,0)*0.10),2)) "Sueldo Neto"  
FROM emp;
```

6. Hacer un listado de empleados con sueldo superior a 1000

```
SELECT ename nombre, sal salario  
FROM emp  
WHERE sal>1000;
```

7. Hacer un listado de empleados (nombre, oficio, sueldo, comisión) que tengan oficio igual a 'CLERK'

```
SELECT ename nombre, job oficio, sal salario, nvl(comm,0) comision  
FROM emp  
WHERE job='CLERK';
```

8. Hacer un listado de empleados (nombre, oficio, sueldo, comisión) que tengan oficio distinto a 'CLERK'

```
SELECT ename nombre, job oficio, sal salario, nvl(comm,0) comision  
FROM emp  
WHERE NOT job='CLERK' [WHERE job<>'CLERK']
```

9. Hacer un listado de empleados (nombre, oficio, sueldo, comisión) que tengan oficio distinto a 'CLERK' y sueldo superior a 1500

```
select ename nombre, job oficio, sal salario, nvl(comm,0) comision  
from emp  
where job<>'CLERK'  
and sal>1500;
```

10. Hacer un listado de empleados (nombre, oficio, sueldo, comisión) que tengan asignada comisión

```
SELECT ename nombre, job oficio, sal salario, comm comision  
FROM emp  
WHERE comm IS NOT NULL;
```

11. Hacer un listado de empleados (nombre, oficio, sueldo, comisión) que NO tengan asignada comisión

```
SELECT ename nombre, job oficio, sal salario, comm comision  
FROM emp  
WHERE comm IS NULL;
```

12. Seleccionar los empleados cuyo nombre empieza por 'A'

```
SELECT ename nombre  
FROM emp  
WHERE ename LIKE 'A%';
```

-
- 13. Seleccionar los empleados cuyo nombre tienen una A en cualquier posición**
SELECT ename nombre
FROM emp
WHERE ename LIKE '%A%';
-
- 14. Seleccionar los empleados cuyo nombre no contiene ninguna 'A'**
SELECT ename nombre
FROM emp
WHERE ename NOT LIKE '%A%';
-
- 15. Seleccionar los empleados cuyo nombre empieza por una vocal**
SELECT ename nombre
FROM emp
WHERE ename LIKE 'A%'
OR ename LIKE 'E%'
OR ename LIKE 'I%'
OR ename LIKE 'O%'
OR ename LIKE 'U%';
-
- 16. Seleccionar los empleados con sueldo entre 1000 y 2000 (ambos inclusive)**
SELECT ename nombre, sal salario
FROM emp
WHERE sal BETWEEN 1000 AND 2000;
-
- 17. Seleccionar los empleados con sueldo igual a 1000, 2000, 3000, 4000 ó 5000**
SELECT ename nombre, sal salario
FROM emp
WHERE sal IN(1000,2000,3000,4000,5000);
O tambien
SELECT ename nombre, sal salario
FROM emp
WHERE sal=1000
OR sal=2000
OR sal=3000
OR sal=4000
OR sal=5000;
-
- 18. Seleccionar los empleados cuyo nombre comienza por A,B,C,J,K,M**
SELECT ename nombre, sal salario
FROM emp
WHERE SUBSTR(ename,1,1) IN ('A','B','C','J','K','M');
-
- 19. Seleccionar los empleados cuyo sueldo es la $5000/2 + 500$ ó $5000-1000$**
SELECT ename nombre, sal salario
FROM emp
WHERE sal IN(5000/2+500,5000-1000);
-
- 20. Seleccionar los empleados cuyo sueldo es la $5000/2 + 500$, $5000-1000$ ó el salario máximo de todos los empleados**
SELECT ename nombre, sal salario
FROM emp
WHERE sal IN(5000/2+500,5000-1000, (SELECT MAX(sal) FROM emp));
-
- 21. Seleccionar nombre, sueldo y sueldo formateado de todos los empleados**
SELECT ename, sal, TO_CHAR(sal, '999,999.99')
FROM emp;
-
- 22. Seleccionar nombre, sueldo y sueldo formateado con el simbolo 'Dólar' de todos los empleados**
SELECT ename, sal, TO_CHAR(sal, '\$99,999.99')
FROM emp;
-
- 23. Seleccionar nombre, sueldo y sueldo formateado con el simbolo 'Moneda local' de todos los empleados**
SELECT ename, sal, TO_CHAR(sal, 'L99,999.99')
FROM emp;
-
- 24. Seleccionar la fecha del sistema (día, mes, año, horas (24), minutos, segundos)**
SELECT TO_CHAR(sysdate, 'dd/mm/yyyy HH24:MI:SS')
FROM dual;
-
- 25. Seleccionar la fecha del sistema (nombre día, día, mes, año, horas (24), minutos, segundos)**
SELECT TO_CHAR(sysdate, 'DAY, dd month yyyy HH24:MI:SS')

FROM dual;

26. Seleccionar la fecha del día 1 de enero de 2005 , mediante una tira de caracteres y su máscara de formato

```
SELECT TO_DATE('01012005','ddmmyyyy')
FROM dual;
```

27. Calcular el número de días vividos por una persona nacida el día 3 de julio de 1970:

```
SELECT ROUND(SYSDATE-TO_DATE('03071970','ddmmyyyy')) "Dias vividos"
FROM dual;
```

28. Calcular el número de segundos transcurridos desde la última medianoche

```
SELECT TO_CHAR(SYSDATE,'sssss')
FROM dual
```

29. Calcular el número horas completas transcurrido desde la última medianoche

```
SELECT TRUNC(TO_NUMBER(TO_CHAR(sysdate,'sssss'))/3600) Horas
FROM dual
```

30. Calcular el número de meses transcurridos entre la fecha de contratación de cada empleado y hoy

```
SELECT ename nombre, MONTHS_BETWEEN(sysdate, hiredate) meses
FROM emp;
```

31. Calcular el último día del mes (fecha) del mes actual

```
SELECT LAST_DAY(SYSDATE)
FROM dual.
```

32. Calcular el último día del mes (fecha) del mes actual, con horas, minutos y segundos

```
SELECT TO_CHAR(LAST_DAY(SYSDATE),'dd/mm/yyy HH24:MI:SS')
FROM dual;
```

33. Calcular en qué MES (cifras) se ha contratado cada empleado

```
SELECT ename nombre, hiredate "Fecha contrato", TO_CHAR(hiredate,'MM') "Mes contrato"
FROM emp;
```

34. Calcular cuanto debería haber cobrado cada empleado en su primer año de trabajo (desde la fecha de contrato hasta el 12 de diciembre de ese año)

```
SELECT ename nombre,
       TO_DATE('3112'||TO_CHAR(hiredate,'yyyy')) "31 Dic",
       TO_DATE('3112'||TO_CHAR(hiredate,'yyyy')-hiredate) "Dias 1er año",
       ((TO_DATE('3112'||TO_CHAR(hiredate,'yyyy')-hiredate)/30)*sal "Devengado 1er año"
FROM emp;
```

35. Cuantos oficios distintos hay en la tabla de empleados

```
SELECT DISTINCT job
FROM emp;
```

36. Calcular el IRPF de cada empleado, teniendo en cuenta que para los 'CLERK se les retiene un 15%, y a los 'ANAYST' un 20%. Al resto se les retiene un 19%.

```
SELECT ename, sal, job, DECODE(job, 'CLERK', sal*0.15, 'ANALYST', sal*0.20, sal*0.19) IRPF
FROM emp;
```

37. Efectuar una propuesta de aumento salarial: Para los empleados del Dept. 10 un 5%, Dept. 20 un 7% , Dept 30 un 8% y al resto un 3% del salario

```
SELECT ename, deptno, sal,
       DECODE(deptno, 10, sal*0.05, 20, sal*0.07, 30, sal*0.08, sal*0.03) "Aumento mensual"
FROM emp;
```

38. Listar los nombres de los empleados, identificando como 'Vendedor' si cobra comisión, y 'No vendedor' si no la cobra

```
SELECT ename nombre, DECODE(comm, NULL, 'NO', 'SI') "Vendedor?"
FROM emp;
```

39. Calcular cuanto se paga mensualmente a todos los empleados

```
SELECT SUM(sal) "Salario total"
FROM emp;
```

40. Calcular cuantos empleados hay en la tabla

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

41. Calcular el sueldo medio de todos los empleados

```
SELECT SUM(sal)/COUNT(*) "Salario medio"
FROM emp;
```

42. Calcular la comision media de todos los empleados

```
SELECT SUM(comm)/COUNT(*) "Comision media"
FROM emp;
```

Comparar con:

```
SELECT SUM(comm)/COUNT(comm) "Comision media"
FROM emp;
```

43. Calcular la comisión media de los empleados

```
SELECT AVG(comm) media
FROM emp;
```

44. Calcular la suma de los sueldos de los empleados del Departamento 20

```
SELECT SUM(sal) "Salarios Dpt 20"
FROM emp
WHERE deptno=20;
```

45. Calcular el sueldo medio de los empleados que pertenezcan al Dept 10 ó 30

```
SELECT AVG(sal) media
FROM emp
WHERE deptno=10
   OR deptno=30;
```

O también:

```
SELECT AVG(sal) media
FROM emp
WHERE deptno IN(10,30);
```

46. Calcular la suma de sueldos que se pagan en cada uno de los departamentos 10 y 30 (por separado)

```
SELECT deptno, SUM(sal) "Salario total"
FROM emp
WHERE deptno IN(10,30)
GROUP BY deptno;
```

47. Calcular cuantos empleados se han contratado cada año

```
SELECT TO_CHAR(hiredate,'YYYY') año,COUNT(*) empleados
FROM emp
GROUP BY TO_CHAR(hiredate,'YYYY');
```

48. Calcular el sueldo máximo y mínimo de cada departamento

```
SELECT deptno, MAX(sal) "Salario max", MIN(sal) "Salario mín"
FROM emp
GROUP BY deptno;
```

49. Calcular cuanto se gana de media por cada oficina. Redondear a 2 decimales

```
SELECT job oficio, ROUND(AVG(sal),2) "Salario medio"
FROM emp
GROUP BY job
```

50. Cuantos días de vacaciones correspondieron a cada empleado el primer año de trabajo (contando 1 día por semana entera trabajada)

```
SELECT ename nombre,
FLOOR((TO_DATE('3112'||TO_CHAR(hiredate,'yyyy'))-hiredate)/7) "dies de vacances"
FROM emp
```

51. Qué día de la semana se contrató a cada empleado

```
SELECT ename nombre, hiredate "Fecha contrato",TO_CHAR(hiredate,'day') "Dia semana"
FROM emp;
```

52. Calcular la paga de beneficios que corresponde a cada empleado (3 salarios mensuales incrementados un: 10% para PRESIDENT, 20% para los MANAGER, 30% para el resto)

```
SELECT ename nombre, job oficio,sal salario,
DECODE(job,'PRESIDENT',sal*3*1.1,'MANAGER',sal*3*1.2,sal*3*1.3) "Paga extra"
FROM emp;
```

O también:

```
SELECT ename nome,job oficio,sal salario,
```

```
sal*3*DECODE(job, 'PRESIDENT', 1.1, 'MANAGER', 1.2, 1.3) "Paga extra"  
FROM emp;
```

53. Cuantos dias han pasado desde el 25 julio de 1992

```
SELECT ROUND(sysdate-TO_DATE('25071992', 'ddmmyyyy')) "Dias desde 25 jul '92"  
FROM dual;
```

54. Seleccionar el producto cartesiano de nombres de empleados y nombres de departamentos

```
SELECT ename, dname  
FROM emp, dept;
```

55. Seleccionar el nombre de cada empleado y el nombre del departamento al que pertenece

```
SELECT ename, dname  
FROM emp, dept  
WHERE emp.deptno=dept.deptno
```

56. Seleccionar el nombre de cada empleado, el nombre del departamento al que pertenece, y el codigo de departamento del empleado

```
SELECT ename, dname, dept.deptno  
FROM emp, dept  
WHERE emp.deptno=dept.deptno
```

57. Seleccionar el nombre de cada empleado, el nombre del departamento al que pertenece, y el codigo de departamento del empleado (ANSI)

```
SELECT ename, dname, deptno  
FROM emp  
NATURAL JOIN dept;
```

58. Seleccionar nombre del empleado y su grado de salario:

```
SELECT ename, grade  
FROM emp, salgrade  
WHERE sal BETWEEN losal AND hisal
```

59. Listar el nombre del empleado y el nombre de su jefe

```
SELECT c.ename, j.ename  
FROM emp c, emp j  
WHERE c.mgr=j.empno
```

60. Listar el nombre del empleado y el nombre de su jefe. Incluir empleados que no tengan jefe

```
SELECT c.ename, j.ename  
FROM emp c, emp j  
WHERE c.mgr=j.empno(+)
```

61. Seleccionar nombre del empleado, nombre del jefe, fechas contrato del trabajador y del jefe, de forma que la fecha de contrato del empleado sea anterior a la de su jefe

```
SELECT c.ename emp, j.ename jefe, c.hiredate "contr. emp", j.hiredate "contr. jefe"  
FROM emp c, emp j  
WHERE c.mgr=j.empno  
AND c.hiredate<j.hiredate;
```

62. Seleccionar nombre del empleado, nombre del jefe, salarios del trabajador y del jefe, de forma que el sueldo del empleado sea inferior a la mitad del salario de su jefe

```
SELECT c.ename empleado, j.ename jefe, c.sal "sal empleado", j.sal "sal jefe"  
FROM emp c, emp j  
WHERE c.mgr=j.empno  
AND c.sal<j.sal/2
```

63. Seleccionar las distintas ubicaciones de los departamentos

```
SELECT DISTINCT loc  
FROM dept
```

64. Seleccionar ubicación, nombre empleado. Incluir ubicaciones de departamentos sin empleados

```
SELECT loc, ename  
FROM emp E, dept D  
WHERE E.deptno(+) = D.deptno
```

65. Seleccionar nombre empleados, y departamento al que pertenecen, para los empleados que ganan menos de 1000\$

```
SELECT ename, dname  
FROM emp E, dept D  
WHERE E.deptno=D.deptno  
AND E.sal<1000
```

66. Seleccionar nombre empleados, y departamento al que pertenecen, para los empleados que ganan menos de 1000\$ (ANSI)

```
SELECT ename, dname  
FROM emp  
NATURAL JOIN dept  
WHERE sal < 1000
```

PRACTICAS SQL (Usuario HR)

67. Conectar como SYSTEM, y desbloquear el usuario HR, asignandole el password HR
ALTER USER HR IDENTIFIED BY HR ACCOUNT UNLOCK;

68. Conectar como HR
CONNECT hr/hr

69. Visualizar las tablas del esquema HR
SELECT *
FROM cat;

70. Consultar las columnas de la tabla EMPLOYEES
DESC employees

71. Listar apellido, nombre, codigo oficio, sueldo de los empleados
SELECT last_name apellido, first_name nombre, job_id "Cod oficio", salary salario
FROM employees;

72. Calcular el sueldo anual que percibe cada empleado
SELECT employee_id, last_name, salary*12 "Salario anual"
FROM employees;

73. Consultar las columnas de la tabla DEPARTMENTS
DESC departments;

74. Listar la tabla de Departamentos
SELECT *
FROM departments;

75. Listar el ID, apellido, cod oficio y fecha contrato de cada empleado
SELECT employee_id ID, last_name apellido, job_id "Cod. cargo", hire_date "Fecha contrato"
FROM employees;

76. Listar los diferentes codigos de oficio de los empleados
SELECT DISTINCT job_id
FROM employees;

77. Listar el ID (Emp#), apellido, cod oficio (Job) y fecha contrato (Hire date) de cada empleado
SELECT employee_id "Emp #", last_name Employee, job_id Job, hire_date "Hire Date"
FROM employees;

78. Listar en una sola columna el nombre y codigo de oficio de cada empleado
SELECT last_name||', '||job_id "Empleado y Cod. oficio"
FROM employees;

79. Listar el id, nombre, apellido, email, teléfono, fecha contrato, cod oficio, sueldo, comision, cod jefe, cod departamento. Ha de ser una columna separada por comas. Llamar a la columna única: '- Salida -'
SELECT employee_id ||', '|| first_name ||', '|| last_name ||', '|| email ||', '||
phone_number ||', '|| hire_date ||', '|| job_id ||', '|| salary ||', '||
commission_pct ||', '|| manager_id ||', '|| department_id "-Salida -"
FROM employees;

80. Listar el apellido y sueldo de los empleados que ganan más de 2000
SELECT last_name apellido, salary salario
FROM employees
WHERE salary > 2000

81. Listar el apellido y cod departamento del empleado con id=176
SELECT last_name apellido, department_id "Cod dept"
FROM employees
WHERE employee_id=176;

82. Listar el apellido y sueldo de los empleados que NO ganan entre 5000 y 12000
SELECT last_name, salary
FROM employees
WHERE salary NOT BETWEEN 5000 AND 12000;

83. Listar el apellido, cod oficio y fecha contrato de empleados contratados entre 20 febrero 1998 y 1 de mayo de 1998. Ordenar por fecha
SELECT last_name, job_id, hire_date

```
FROM employees
WHERE hire_date BETWEEN '20/02/1998' AND '01/05/1998'
ORDER BY hire_date ASC;
```

84. Listar el apellido y cod departamento de los empleados pertenecientes a los departamentos 20 o 50. Ordenar por apellido

```
SELECT last_name, department_id
FROM employees
WHERE department_id IN(20,50)
ORDER BY last_name;
```

85. Listar el apellido (alias: Empleado) y sueldo (alias: Salario mensual) de los empleados pertenecientes a los departamentos 20 o 50, que ganen entre 5000 y 12000.

```
SELECT last_name "Empleado", salary "Salario mensual"
FROM employees
WHERE salary BETWEEN 5000 AND 12000
AND department_id IN (20,50);
```

86. Listar el apellido y fecha contrato de empleados contratados el año 1994

```
SELECT last_name, hire_date
FROM employees
WHERE TO_CHAR(hire_date, 'yyyy') = '1994';
```

87. Listar el apellido, cod oficio de empleados sin jefe asignado

```
SELECT last_name, job_id
FROM employees
WHERE manager_id IS NULL;
```

88. Listar el apellido, sueldo y % comisión de los empleados que cobran comisión. Ordenar por salario y comisión, de forma que los más altos salgan primero

```
SELECT last_name, salary, commission_pct
FROM employees
WHERE commission_pct IS NOT NULL
ORDER BY salary DESC, commission_pct DESC;
```

89. Listar el apellido de los empleados que tengan una 'a' en la 3a letra del apellido

```
SELECT last_name
FROM employees
WHERE last_name LIKE '__a%';
```

90. Listar el apellido de los empleados que tengan una 'a' y una 'e' en el apellido

```
SELECT last_name
FROM employees
WHERE last_name LIKE '%a%'
AND last_name LIKE '%e%';
```

91. Listar el apellido, cod oficio, sueldo de empleados 'CLERK' o 'SA_REP', y con salario distinto de: 2500, 3500, 7000

```
SELECT last_name, job_id, salary
FROM employees
WHERE (job_id = 'ST_CLERK' OR job_id = 'SA_REP')
AND salary NOT IN(2500,3500,7000);
```

92. Listar el apellido (alias: Empleado), sueldo (alias: Salario mensual) y % comisión de los empleados cuyo % comisión sea del 20%

```
SELECT last_name "Empleado", salary "Salario mensual", commission_pct
FROM employees
WHERE commission_pct=0.2;
```

93. Listar el apellido y salario de los empleados, de forma que estén separados por una línea de puntos, y que de extremo a extremo haya 30 caracteres

```
SELECT RPAD(last_name,20, '.') || LPAD(salary,10, '.')
FROM employees;
```

94. Listar los apellidos de los empleados, reemplazando la letra 'a' por un '1'

```
SELECT REPLACE(last_name, 'a', '1')
FROM employees;
```

95. Obtener la fecha del sistema (alias: Fecha) en el formato por defecto

```
SELECT sysdate "Fecha"
FROM dual;
```


96. Listar el id, apellido y salario de los empleados, y el salario incrementado un 15%, redondeado (alias: Nuevo salario)

```
SELECT employee_id, last_name, salary, ROUND(salary*1.15) "Nuevo salario"
FROM employees;
```

97. Listar el id, apellido y salario de los empleados, el salario incrementado un 15%, redondeado (alias: Nuevo salario) y la cuantía del incremento (alias: Incremento)

```
SELECT employee_id, last_name, salary,
       ROUND(salary*1.15) "Nuevo salario",
       ROUND(salary*0.15) "Incremento"
FROM employees;
```

98. Listar el apellido y longitud del apellido de los empleados cuyo nombre empiece por A, J ó M. Ordenar por apellido. (Los apellidos tendrán la inicial en mayúscula y el resto en minúsculas)

```
SELECT INITCAP(last_name) Name, length(last_name) Length
FROM employees
WHERE last_name LIKE 'A%'
OR last_name LIKE 'J%'
OR last_name LIKE 'M%'
ORDER BY last_name;
```

99. Listar el apellido número de meses trabajados, redondeando al entero superior. Ordenar por número de meses trabajados, de menor a mayor

```
SELECT last_name, CEIL(MONTHS_BETWEEN(sysdate, hire_date)) MESES
FROM employees
ORDER BY 2 ;
```

100. Componer una frase con el apellido, salario actual y el triple del salario de los empleados. Por ejemplo: 'Pepe gana 1000 mensuales, pero le gustaria ganar 3000'. (alias: Sueldos soñados)

```
SELECT last_name||' gana '||salary|| ' mensuales, pero quería ganar '||salary*3 "Sueldos soñados"
FROM employees;
```

101. Listar el apellido y salario de los empleados, rellenando por la izquierda hasta 15 caracteres con el símbolo '\$'

```
SELECT last_name, LPAD(salary,15,'$') salary
FROM employees;
```

102. Listar apellido, fecha contrato y fecha revision contrato de los empleados, sabiendo que será el lunes siguiente a la fecha que cumple 6 meses de trabajo en la empresa. Formatear esta fecha de acuerdo con el ejemplo: 'Lunes, 12th de noviembre de 2005'

```
SELECT last_name, hire_date,
       TO_CHAR(NEXT_DAY((ADD_MONTHS(hire_date,6)),1), 'DAY ", " ddspth " de " MONTH YYYY') Rev
FROM employees;
```

103. Listar apellido, fecha contrato y día de la semana en que fueron contratados los empleados, ordenando por día de la semana, de forma que queden ordenados por Lunes, Martes, Miercoles, Jueves , Viernes, Sabado, Domingo

```
SELECT last_name, hire_date, TO_CHAR(hire_date, 'DAY') "Dia sem"
FROM employees
ORDER BY TO_CHAR(hire_date, 'D');
```

104. Listar apellido y comisión de los empleados. En el caso de que no gane comisión, sacar la frase 'Sin comision'

```
SELECT last_name, NVL(TO_CHAR(commission_pct), 'Sin comision')
FROM employees;
```

105. Listar apellido de los empleados, y una tira de asteriscos, de forma que hay un asterisco por cada 1000\$ (redondeado) que gana el empleado. Titula la columna 'Empleado y su salario'. Ordenar esta columna de forma que los que más ganan aparezcan primero. Los apellidos deben quedar ajustados a la longitud del apellido más largo.

```
SELECT RPAD(last_name, (select max(length(last_name)) from employees), '.')
||': '||RPAD(' ', ROUND(salary/1000), '*') "Empleado y su salario"
FROM employees
ORDER BY salary DESC
```

106. Listar los distintos codigos de oficio de los empleados, de forma que : AD_PRES se traduzca por A, ST_MAN por B, IT_PROG por C, SA_REP por D y el resto por 0

```
SELECT DISTINCT job_id,
       DECODE(job_id, 'AD_PRES', 'A', 'ST_MAN', 'B', 'IT_PROG', 'C', 'SA_REP', 'D', 'ST_CLERK', 'E', 0)
```

from employees

107. Listar la tabla LOCATIONS

```
SELECT *
FROM locations;
```

108. Seleccionar el nombre de departamento, y el nombre de la ciudad donde está

```
SELECT d.department_id,l.city
FROM departments d,locations l
WHERE d.location_id=l.location_id;
```

109. Listar las diferentes ciudades de la tabla LOCATIONS

```
SELECT DISTINCT city
FROM locations;
```

110. Listar el nombre de departamento y nombre de ciudad para los departamento radicados en Seattle

```
SELECT department_name,department_id
FROM departments D, locations L
WHERE D.location_id=L.location_id
AND L.city='Seattle';
```

111. Seleccionar el nombre de países y la region a que pertenecen (ANSI)

```
SELECT country_name, region_name
FROM countries
NATURAL JOIN regions
```

112. Seleccionar el nombre de países y la region a que pertenecen

```
SELECT country_name, region_name
FROM countries C,regions R
WHERE C.region_id=R.region_id
```

113. Seleccionar apellido de los empleados y nombre de la ciudad donde está el departamento a que pertenecen

```
SELECT E.last_name nombre,L.city ciudad
FROM employees E, locations L, departments D
WHERE D.department_id=E.department_id
AND D.location_id=L.location_id
```

114. Seleccionar nombre ciudades y nombre departamentos radicados en Canadá

```
SELECT L.city ciudad,D.department_name department
FROM locations L,departments D
WHERE L.location_id=D.location_id
AND L.country_id='CA'
```

O también

```
SELECT L.city ciudad,D.department_name department
FROM locations L,departments D, countries C
WHERE L.location_id=D.location_id
AND L.country_id=C.country_id
AND C.country_name='Canada'
```

115. Listar nombre completo de los trabajadores, oficio, nombre departamento y ciudad de aquellos que no son programadores ('Programmer')

```
SELECT first_name||' '||last_name nombre,job_title oficio,
       department_name departamento,city ciudad
FROM employees E,jobs J,departments D, locations L
WHERE J.job_id = E.job_id
AND D.department_id = E.department_id
AND L.location_id = D.location_id
AND job_title <> 'Programmer'
```

116. Listar nombre completo de los trabajadores, oficio, nombre departamento y ciudad de aquellos que no son programadores ('Programmer') (ANSI)

```
SELECT first_name||' '||last_name nombre,job_title oficio,
       department_name departamento,city ciudad
FROM employees
NATURAL JOIN departments
NATURAL JOIN locations
NATURAL JOIN jobs
WHERE job_title<>'Programmer'
```

117. Seleccionar nombre de departamento, ciudad, codigo postal y estado. Si no hay estado, que salga 'Desconocido'

```
SELECT department_name "Departamento",city "Ciudad",postal_code "Codigo postal",
       NVL(state_province,'Desconocido') estado
FROM departments D,locations L
WHERE D.location_id=L.location_id
```

118. Seleccionar nombre de departamento, ciudad, codigo postal y estado. Si no hay estado, que salga 'Desconocido' (ANSI)

```
SELECT department_name departamento,city ciudad,postal_code "codigo postal",
       NVL(state_province,'Desconocida') provincia
FROM locations
NATURAL JOIN departments
```

119. Seleccionar apellido,cod departamento y nombre departamento al que pertenecen los empleados

```
SELECT last_name apellido, D.department_id " Id depart.", department_name departamento
FROM employees E, departments D
WHERE E.department_id = D.department_id
```

120. Seleccionar apellido,cod departamento y nombre departamento al que pertenecen los empleados (ANSI)

```
SELECT last_name apellido,department_id "Id depart.", department_name departamento
FROM employees
NATURAL JOIN departments
```

121. Seleccionar las distintas combinaciones cod oficio, cod ubicacion para los empleados del departamento cod 80

```
SELECT DISTINCT job_id,location_id
FROM departments D, employees E
WHERE E.department_id=D.department_id
AND E.department_id=80;
```

122. Seleccionar las distintas combinaciones cod oficio, cod ubicacion para los empleados del departamento cod 80 (ANSI)

```
SELECT DISTINCT job_id,location_id
FROM locations
NATURAL JOIN departments D
JOIN employees E ON (D.department_id = E.department_id)
WHERE E.department_id = 80;
```

123. Seleccionar apellido, nombre departamento, cod ubicacion, ciudad para los empleados que ganan comisión

```
SELECT last_name, department_name, D.location_id, L.city
FROM employees E,departments D, locations L
WHERE E.department_id = D.department_id
AND D.location_id = L.location_id
AND commission_pct IS NOT NULL
```

124. Seleccionar apellido, nombre departamento, cod ubicacion, ciudad para los empleados que ganan comisión (ANSI)

```
SELECT last_name,department_name,location_id, city
FROM locations
NATURAL JOIN departments
JOIN employees USING (department_id)
WHERE employees.commission_pct IS NOT NULL
```

125. Seleccionar apellido, nombre departamento para los empleados que tengan una 'a' en su apellido

```
SELECT last_name, department_name
FROM employees E,departments D
WHERE E.department_id = D.department_id
AND last_name LIKE '%a%'
```

126. Seleccionar apellido, nombre departamento para los empleados que tengan una 'a' en su apellido (ANSI)

```
SELECT last_name, department_name
FROM employees
JOIN departments USING (department_id)
WHERE last_name LIKE '%a%'
```

127. Seleccionar apellido, cod oficina, cod departamento, nombre departamento de los empleados cuyo departamento está radicado en 'toronto' (Todo en minúsculas)

```
SELECT last_name, job_id, e.department_id, department_name
FROM locations l, departments d, employees e
WHERE d.department_id=e.department_id
AND l.location_id=d.location_id
AND LOWER (l.city)='toronto';
```

128. Seleccionar apellido, cod oficina, cod departamento, nombre departamento de los empleados cuyo departamento está radicado en 'toronto' (Todo en minúsculas) (ANSI). Empezar por tabla DEPARTMENTS

```
SELECT last_name, job_id, department_id, department_name
FROM departments
NATURAL JOIN locations
JOIN employees USING (department_id)
WHERE LOWER(locations.city) = 'toronto'
```

129. Seleccionar apellido, cod oficina, cod departamento, nombre departamento de los empleados cuyo departamento está radicado en 'toronto' (Todo en minúsculas) (ANSI). Empezar por tabla EMPLOYEES

```
SELECT last_name, job_id, department_id, department_name
FROM employees
JOIN departments USING (department_id)
JOIN locations USING (location_id)
WHERE LOWER(city)='toronto'
```

130. Seleccionar apellido, cod empleado, apellido jefe, cod empleado jefe de todos los empleados

```
SELECT E.last_name "Empleado", E.employee_id "Id empl",
       J.last_name "Jefe", J.employee_id "Id jefe"
FROM employees E, employees J
WHERE J.employee_id = E.manager_id
```

131. Seleccionar apellido, cod empleado, apellido jefe, cod empleado jefe de todos los empleados (ANSI)

```
SELECT E.last_name "Empleado", E.employee_id "Id empl",
       J.last_name "Jefe", J.employee_id "Id jefe"
FROM employees E
JOIN employees J ON ( J.employee_id = E.manager_id )
```

132. Seleccionar apellido, cod empleado, apellido jefe, cod empleado jefe de todos los empleados, y que salgan incluso los empleados sin jefe. Ordenar por cod empleado. (ANSI)

```
SELECT E.last_name "Empleado", E.employee_id "Id empl",
       J.last_name "Jefe", J.employee_id "Id jefe"
FROM employees E
LEFT OUTER JOIN employees J ON ( J.employee_id = E.manager_id )
ORDER BY e.employee_id
```

O también:

```
SELECT E.last_name "Empleado", E.employee_id "Id empl",
       J.last_name "Jefe", J.employee_id "Id jefe"
FROM employees E
JOIN employees J ON ( J.employee_id(+) = E.manager_id )
ORDER BY E.employee_id
```

133. Seleccionar cod departamento, apellido empleado y apellido colega (colega: trabaja en el mismo departamento). Evitar que un empleado salga como colega de sí mismo

```
SELECT E.department_id "ID depart", E.last_name "Empleado", C.last_name "Colega"
FROM employees E, employees C
WHERE E.department_id = C.department_id
AND E.employee_id <> C.employee_id;
```

134. Seleccionar cod departamento, apellido empleado y apellido colega (colega: trabaja en el mismo departamento). Evitar que un empleado salga como colega de sí mismo (ANSI)

```
SELECT E.department_id "ID depart", E.last_name "Empleado", C.last_name "Colega"
FROM employees E
JOIN employees C ON (E.department_id = C.department_id)
WHERE E.employee_id <> C.employee_id;
```

135. Seleccionar apellido y fecha contrato de los empleados contratados posteriormente al empleado 'Davies'

```
SELECT E.last_name, E.hire_date
FROM employees E, employees D
WHERE D.last_name = 'Davies'
AND E.hire_date > D.hire_date
```

136. Seleccionar apellido y fecha contrato de los empleados contratados posteriormente al empleado 'Davies' (ANSI)

```
SELECT E.last_name, E.hire_date
FROM employees E
JOIN employees D ON (D.last_name = 'Davies')
WHERE e.hire_date>d.hire_date
```

137. Seleccionar apellido y fecha de contrato de los empleados, apellido y fecha contarto de su jefe, de tal forma que solo salgan los empleados cuyo contrato fue anterior al de su jefe. (ANSI)

```
SELECT e.last_name Emp,e.hire_date "Contr emp",m.last_name Jefe,m.hire_date "Contr jefe"
FROM employees E
JOIN employees J
ON E.manager_id = J.employee_id
WHERE E.hire_date < J.hire_date
```

138. Listar todas las posibles combinaciones de nombres y apellidos de empleados (ANSI)

```
SELECT a.first_name, b.last_name
FROM employees a
CROSS JOIN employees b
```

139. Listar todas las posibles combinaciones de nombres y apellidos de empleados, de forma que las iniciales del nombre y el apellido sean diferentes (ANSI)

```
SELECT a.first_name, b.last_name
FROM employees a
CROSS JOIN employees b
WHERE SUBSTR(a.first_name,1,1)<>SUBSTR(b.first_name,1,1)
```

140. Listar todas las posibles combinaciones de nombres y apellidos de empleados, de forma que la longitud del nombre sea menor que la del apellido (ANSI)

```
SELECT a.first_name, b.last_name
FROM employees a
CROSS JOIN employees b
WHERE LENGTH(a.first_name) < LENGTH(b.last_name)
```

141. Seleccionar nombre de empleados, departamento al que pertenece, para los departamentos que empiecen con 'S' (ANSI)

```
SELECT last_name, department_name
FROM employees
JOIN departments USING (department_id)
WHERE department_name LIKE 'S%'
```

142. Seleccionar ciudades y pais al que pertenece la ciudad (ANSI)

```
SELECT city,country_name
FROM locations
NATURAL JOIN countries
```

143. Seleccionar paises y ciudades donde radican deparatamentos. Incluir paises que no haya departamento (ANSI)

```
SELECT country_name, city
FROM countries
LEFT OUTER JOIN locations USING(country_id)
```

O también:

```
SELECT country_name, city
FROM locations
RIGHT OUTER JOIN countries USING(country_id)
```

144. Seleccionar paises y ciudades donde radican deparatamentos. Incluir paises que no haya departamento Si no hay ciudad, scar la frase 'No existe ciudad'. (ANSI)

```
SELECT country_name, NVL(city,'No existe ciudad') ciudad
FROM locations
RIGHT OUTER JOIN countries USING(country_id)
```

145. Calcular el salario medio por nombre de departamento (ANSI)

```
SELECT department_name,AVG(salary) Media
FROM employees
JOIN departments USING (department_id)
GROUP BY department_name;
```

146. Calcular el salario medio por nombre de departamento. Incluir los departamentos sin empleados asignados (ANSI)

```
SELECT department_name,AVG(salary)
FROM employees
LEFT OUTER JOIN departments USING (department_id)
GROUP BY department_name
```

147. Calcular el salario máximo, mínimo, medio y la suma de salarios de todos los empleados. Redondear las cantidades

```
SELECT
    ROUND(MAX(salary)) Maximo,
    ROUND(MIN(salary)) Minimo,
    ROUND(AVG(salary)) Media,
    ROUND(SUM(salary)) Suma
FROM employees;
```

148. Calcular el salario máximo, mínimo, medio y la suma de salarios de todos los empleados, desglosado por cod oficio. Redondear las cantidades

```
SELECT job_id,
    ROUND(MAX(salary)) Maximo,
    ROUND(MIN(salary)) Minimo,
    ROUND(AVG(salary)) Media,
    ROUND(SUM(salary)) Suma
FROM employees
GROUP BY job_id;
```

149. Calcular cuantos empleados hay para cada cod oficio

```
SELECT job_id cargo, COUNT (employee_id) Num_EmpIs
FROM employees
GROUP BY job_id;
```

150. Calcular cuantos mandos hay en la empresa. (Mando: aquel que figura en la columna manager_id de algún empleado)

```
SELECT COUNT (DISTINCT manager_id) "Num de Mandos"
FROM employees;
```

151. Calcular la diferencia entre el mayor sueldo de la empresa y el menor.

```
SELECT MAX(salary) - MIN(salary) Diferencia
FROM employees;
```

152. Calcular el salario mínimo de los empleados de cada mando intermedio. Excluir los empleados sin jefe asignado. Evitar que salgan los grupos con salario mínimo menor de 6.000\$. Ordenar en sentido descendente

```
SELECT manager_id, MIN(salary)
FROM employees
WHERE salary<=6000
AND manager_id IS NOT NULL
GROUP BY manager_id
ORDER BY MIN(salary) DESC
```

153. Calcular cuantos empleados trabajan por nombre de departamento y codigo de ubicación, así como el sueldo medio redondeado

```
SELECT
    department_name "Departamento",
    location_id "ID ubic",
    COUNT(employee_id) "Num. Empls",
    ROUND(AVG(salary),2) "Salario"
FROM employees E, departments D
WHERE E.department_id = D.department_id
GROUP BY department_name, location_id
```

154. Calcular cuantos empleados trabajan por nombre de departamento y codigo de ubicación, así como el sueldo medio redondeado (ANSI)

```
SELECT
    department_name "Departamento",
    location_id "ID ubic",
    COUNT(employee_id) "Num. Empls",
    ROUND(AVG(salary),2) "Salario"
FROM employees
JOIN departments USING(department_id)
GROUP BY department_name,location_id
```

155. Listar (horizontalmente) cuantos empleados hay, cuantos se contrataron en: 1995,1996,1997,1998

```
SELECT COUNT(*) Total,
SUM (DECODE(TO_CHAR(hire_date,'YYYY'),1995,1,0)) "1995",
SUM (DECODE(TO_CHAR(hire_date,'YYYY'),1996,1,0)) "1996",
SUM (DECODE(TO_CHAR(hire_date,'YYYY'),1997,1,0)) "1997",
SUM (DECODE(TO_CHAR(hire_date,'YYYY'),1998,1,0)) "1998"
FROM employees
```

156. Listar en forma de tabla cuanto se paa de salario por cod oficio (vertical) y de los departamentos 20,50, 80 y 90 (horizontal). Ordenar por cod oficio

```
SELECT job_id Cargo,
SUM(DECODE(department_id,20,salary,null)) "dept 20",
SUM(DECODE(department_id,50,salary,null)) "dept 50",
SUM(DECODE(department_id,80,salary,null)) "dept 80",
SUM(DECODE(department_id,90,salary,null)) "dept 90",
SUM(salary) Total
FROM employees
GROUP BY job_id
ORDER BY job_id
```

157. Calcular los empleados contratados por cod oficio y año de contratación (solo para los cod oficio: ST_CLERK,SH_CLERCK,IT_PROG,HR_REP)

```
SELECT job_id, TO_CHAR(hire_date,'YYYY'),COUNT(*)
FROM employees
WHERE job_id IN ('ST_CLERK','SH_CLERCK','IT_PROG','HR_REP')
GROUP BY job_id, TO_CHAR(hire_date,'YYYY')
```

158. Calcular los empleados contratados por año y cod oficio (solo para los cod oficio: ST_CLERK,SH_CLERCK,IT_PROG,HR_REP). Ordenar por cod oficio.

```
SELECT TO_CHAR(hire_date,'YYYY') AÑO,
SUM(DECODE(job_id,'ST_CLERK',1,0)) ST_CLERK,
SUM(DECODE(job_id,'SH_CLERK',1,0)) SH_CLERK,
SUM(DECODE(job_id,'IT_PROG',1,0)) IT_PROG,
SUM(DECODE(job_id,'HR_REP',1,0)) HR_REP
FROM employees
GROUP BY TO_CHAR(hire_date,'YYYY')
ORDER BY TO_CHAR(hire_date,'YYYY')
```

159. Calcular cuantos empleados hay por cod departamento (vertical) y cod oficio (horizontal). (solo para los cod oficio: ST_CLERK,SH_CLERCK,IT_PROG,HR_REP)

```
SELECT department_id "Cod.Departamento",
SUM(DECODE(job_id,'ST_CLERK',1,0)) ST_CLERK,
SUM(DECODE(job_id,'SH_CLERK',1,0)) SH_CLERK,
SUM(DECODE(job_id,'IT_PROG',1,0)) IT_PROG,
SUM(DECODE(job_id,'HR_REP',1,0)) HR_REP
FROM employees
GROUP BY department_id
```

160. Calcular cuantos empleados hay por cod departamento (vertical) y cod oficio (horizontal). (solo para los cod oficio: ST_CLERK,SH_CLERCK,IT_PROG,HR_REP). Eliminar las filas donde no haya datos, debido a empleados sin departamento.

```
SELECT department_id "Cod.Departamento",
SUM(DECODE(job_id,'ST_CLERK',1,0)) ST_CLERK,
SUM(DECODE(job_id,'SH_CLERK',1,0)) SH_CLERK,
SUM(DECODE(job_id,'IT_PROG',1,0)) IT_PROG,
SUM(DECODE(job_id,'HR_REP',1,0)) HR_REP
FROM employees
WHERE department_id IS NOT NULL
GROUP BY department_id
```

161. Calcular cuantos empleados hay por cod departamento (vertical) y cod oficio (horizontal). (solo para los cod oficio: ST_CLERK,SH_CLERCK,IT_PROG,HR_REP). Eliminar las filas donde no haya datos, debido a empleados sin departamento. (ANSI)

```
SELECT department_id "Cod.Departamento",
SUM(DECODE(job_id,'ST_CLERK',1,0)) ST_CLERK,
SUM(DECODE(job_id,'SH_CLERK',1,0)) SH_CLERK,
SUM(DECODE(job_id,'IT_PROG',1,0)) IT_PROG,
SUM(DECODE(job_id,'HR_REP',1,0)) HR_REP
FROM employees
```

```
JOIN departments USING (department_id)
GROUP BY department_id
```

162. Calcular el número de empleados contratados por año (vertical) y mes (horizontal). Ordenar por año

```
SELECT TO_CHAR(hire_date, 'YYYY') AÑO,
SUM(DECODE(TO_CHAR(hire_date, 'MM'), '01', 1, 0)) ene,
SUM(DECODE(TO_CHAR(hire_date, 'MM'), '02', 1, 0)) feb,
SUM(DECODE(TO_CHAR(hire_date, 'MM'), '03', 1, 0)) mar,
SUM(DECODE(TO_CHAR(hire_date, 'MM'), '04', 1, 0)) abr,
SUM(DECODE(TO_CHAR(hire_date, 'MM'), '05', 1, 0)) may,
SUM(DECODE(TO_CHAR(hire_date, 'MM'), '06', 1, 0)) jun,
SUM(DECODE(TO_CHAR(hire_date, 'MM'), '07', 1, 0)) jul,
SUM(DECODE(TO_CHAR(hire_date, 'MM'), '08', 1, 0)) ago,
SUM(DECODE(TO_CHAR(hire_date, 'MM'), '09', 1, 0)) sep,
SUM(DECODE(TO_CHAR(hire_date, 'MM'), '10', 1, 0)) oct,
SUM(DECODE(TO_CHAR(hire_date, 'MM'), '11', 1, 0)) nov,
SUM(DECODE(TO_CHAR(hire_date, 'MM'), '12', 1, 0)) dic
FROM employees
GROUP BY TO_CHAR(hire_date, 'YYYY')
ORDER BY TO_CHAR(hire_date, 'YYYY')
```

163. Determinar cuanto paga cada departamento por cod oficio. Hacerlo para los cod oficio: ST_CLERK, ST_MAN, SH_CLERK, IT_PROG, HR_REP, PR_REP

```
SELECT department_id Departamento,
SUM(DECODE(job_id, 'ST_CLERK', salary, 0)) ST_CLERK,
SUM(DECODE(job_id, 'ST_MAN', salary, 0)) ST_MAN,
SUM(DECODE(job_id, 'SH_CLERK', salary, 0)) SH_CLERK,
SUM(DECODE(job_id, 'IT_PROG', salary, 0)) IT_PROG,
SUM(DECODE(job_id, 'HR_REP', salary, 0)) HR_REP,
SUM(DECODE(job_id, 'PR_REP', salary, 0)) PR_REP
FROM employees
GROUP BY department_id
ORDER BY department_id
```

164. Determinar cuanto paga cada departamento por cod oficio. Hacerlo para los cod oficio: ST_CLERK, ST_MAN, SH_CLERK, IT_PROG, HR_REP, PR_REP. Hacer que no salgan los ceros.

```
SELECT department_id Departamento,
SUM(DECODE(job_id, 'ST_CLERK', salary, NULL)) ST_CLERK,
SUM(DECODE(job_id, 'ST_MAN', salary, NULL)) ST_MAN,
SUM(DECODE(job_id, 'SH_CLERK', salary, NULL)) SH_CLERK,
SUM(DECODE(job_id, 'IT_PROG', salary, NULL)) IT_PROG,
SUM(DECODE(job_id, 'HR_REP', salary, NULL)) HR_REP,
SUM(DECODE(job_id, 'PR_REP', salary, NULL)) PR_REP
FROM employees
WHERE department_id IS NOT NULL (*)
GROUP BY department_id
ORDER BY department_id
```

(*) He afegit aquesta condició perquè tenim una fila sense nº de departmanet i sense cap salari.

165. Averiguar cuantos empleados residen en cada ciudad, por cada cod oficio. Las idades son: Seattle, Toronto, Oxford, Munich

```
SELECT job_title,
SUM(DECODE(city, 'Seattle', 1, 0)) Seattle,
SUM(DECODE(city, 'Toronto', 1, 0)) Toronto,
SUM(DECODE(city, 'London', 1, 0)) London,
SUM(DECODE(city, 'Oxford', 1, 0)) Oxford,
SUM(DECODE(city, 'Munich', 1, 0)) Munich
FROM jobs
NATURAL JOIN employees
JOIN departments USING (department_id)
NATURAL JOIN locations
GROUP BY job_title
```

O también:

```
SELECT job_title,
SUM(DECODE(city, 'Seattle', 1, 0)) Seattle,
SUM(DECODE(city, 'Toronto', 1, 0)) Toronto,
```



```

SUM(DECODE(city, 'London', 1, 0)) London,
SUM(DECODE(city, 'Oxford', 1, 0)) Oxford,
SUM(DECODE(city, 'Munich', 1, 0)) Munich
FROM employees
JOIN departments USING (department_id)
JOIN locations USING (location_id)
JOIN jobs USING (job_id)
GROUP BY job_title

```

166. Que cod oficio tienen un solo empleado:

```

SELECT job_id trabajo, COUNT(*)
FROM employees
GROUP BY job_id
HAVING COUNT(*)=1

```

167. Listar Cod Departamentos cuya media de salarios es inferior a 7000

```

SELECT department_id, AVG(salary)
FROM employees
GROUP BY department_id
HAVING AVG(salary) < 7000

```

168. Listar cod departamentos cuya media de salarios es inferior a 7000 (contando sólo los empleados que ganan menos de 4.000\$)

```

SELECT department_id, AVG(salary)
FROM employees
WHERE salary<=4000
GROUP BY department_id
HAVING AVG(salary)<7000

```

169. Encontrar el apellido y salario de empleados que ganan menos de 2.000\$. De este conjunto, seleccionar los que tengan una 'a' en el apellido

```

SELECT * from
  (SELECT last_name, salary
   FROM employees
   WHERE salary>2000)
WHERE last_name LIKE '%a%'

```

170. Encontrar el apellido y salario de empleados que ganan menos de 2.000\$. Poner alias 'nom' y 'sou'. De este conjunto, seleccionar los que tengan una 'a' en el NOM

```

SELECT * from
  (SELECT last_name nom, salary sou
   FROM employees
   WHERE salary>2000)
WHERE nom LIKE '%a%'

```

171. Encontrar el nombre y salario de empleados que gana menos de 2.000\$. Poner alias 'nom' y 'sou'. De este conjunto, cambiar los alias a 'nombre' y 'sueldo'. Seleccionar los que tengan una 'a' en el NOM. De estos, seleccionar el SUELDO

```

SELECT sueldo from
  (SELECT nom nombre, sou sueldo FROM
   (SELECT last_name nom, salary sou
    FROM employees
    WHERE salary>2000)
   WHERE nom LIKE '%a%')

```

172. Seleccionar apellidos y salario de los empleados que ganan mas que la media de empleados del departamento 20

```

SELECT last_name, salary
FROM employees
WHERE salary>
  (SELECT AVG(salary)
   FROM employees
   WHERE department_id=20)

```

173. Listar apellido, sueldo fecha contrato de los empleados contratados después de haber sido contratado el primer trabajador del departamento 30. Ordenar por fecha contrato

```

SELECT last_name, salary, hire_date
FROM employees
WHERE hire_date>

```

```
(SELECT MIN(hire_date)
FROM employees
WHERE department_id=30)
ORDER BY hire_date
```

174. Listar apellido, sueldo fecha contrato de los empleados contratados después de haber sido contratado el primer trabajador del departamento 30 y antes que el último contratado del departamento 30. Ordenar por fecha contrato

```
SELECT last_name, salary, hire_date
FROM employees
WHERE hire_date >
    (SELECT MIN(hire_date)
    FROM employees
    WHERE department_id=30)
AND hire_date <
    (SELECT MAX(hire_date)
    FROM employees
    WHERE department_id=30)
ORDER BY hire_date
```

O tambien

```
SELECT last_name, salary, hire_date
FROM employees
WHERE hire_date BETWEEN
    (SELECT MIN(hire_date)
    FROM employees
    WHERE department_id=30)
AND
    (SELECT MAX(hire_date)
    FROM employees
    WHERE department_id=30)
ORDER BY hire_date
```

175. Listar apellido, sueldo, fecha contrato de todos los empleados que ganan más de la media de su departamento

```
SELECT last_name, salary, hire_date
FROM employees E
WHERE salary >
    (SELECT AVG(salary)
    FROM employees
    WHERE department_id = E.department_id);
```

176. Listar apellido, sueldo, fecha contrato de todos los empleados que ganan entre el menor y mayor sueldo de todos los que tienen de cod oficio SA_REP

```
SELECT last_name nombre, salary salario, hire_date "fecha contrato"
FROM employees
WHERE salary BETWEEN
    (SELECT min_salary
    FROM jobs
    WHERE job_id LIKE 'SA_REP')
AND
    (SELECT max_salary
    FROM jobs
    WHERE job_id LIKE 'SA_REP');
```

177. Listar apellido, sueldo de todos los empleados que ganan un sueldo igual a cualquiera de los que gana un empleado con cod oficio SA_REP

```
SELECT last_name, salary
FROM employees
WHERE salary IN
    (SELECT salary
    FROM employees
    WHERE job_id='SA_REP');
```

178. Listar apellido, sueldo de todos los empleados que ganan un sueldo superior a alguno de los que gana un empleado con cod oficio SA_REP

```
SELECT last_name, salary
FROM employees
WHERE salary > ANY
    (SELECT salary
    FROM employees
    WHERE job_id='SA_REP');
```

179. Listar apellido, sueldo de todos los empleados que ganan un sueldo superior a todos los que ganan los empleados con cod oficio SA_REP

```
SELECT last_name, salary
FROM employees
WHERE salary > ALL
      (SELECT salary
       FROM employees
       WHERE job_id='SA_REP');
```

180. Seleccionar apellido, cod departamento, nombre departamento de los empleados. Hacer que sigan los departamentos sin empleados y los empleados sin departamento asignado

```
SELECT e.last_name, e.department_id,d.department_name
FROM employees E
FULL OUTER JOIN departments D ON E.department_id = D.department_id
```

181. Seleccionar apellido y fecha contrato de los colegas de Zlotkey (Que trabajan en el mismo departamento). Excluir el mismo Zlotkey

```
SELECT last_name Apellido, hire_date "Fecha con."
FROM employees
WHERE department_id=
      (SELECT department_id
       FROM employees
       WHERE last_name='Zlotkey')
AND last_name <> 'Zlotkey';
```

182. Seleccionar ID empleado, apellido y sueldo de los empleados que gan más que la media de la empresa. Ordenar por sueldo

```
SELECT employee_id "Numero emp",last_name Apellido, salary Salario
FROM employees
WHERE salary>
      (SELECT AVG(salary)
       FROM employees)
ORDER BY salary;
```

O también:

```
SELECT employee_id "Numero emp",
last_name Apellido,
salary Salario
FROM employees
WHERE salary>
      (SELECT AVG(salary)
       FROM employees)
ORDER BY 3;
```

183. Seleccionar id de empleado, apellido de todos los colegas de empleados que tengan una 'u' en su apellido. (Colega: que trabaja en el mismo departamento)

```
SELECT employee_id "N° empleado",
last_name Apellido
FROM employees
WHERE department_id IN
      (SELECT department_id
       FROM employees
       WHERE last_name LIKE '%u%')
```

184. Seleccionar apellido, ID de empleado y cod oficio de empleados que trabajan en Departamentos cuyo cod ubicación es el 1700

```
SELECT last_name Apellido, department_ID, job_id
FROM employees
JOIN departments USING (department_id)
WHERE location_id =1700
```

O también:

```
SELECT last_name Apellido, department_ID, job_id
FROM employees
WHERE department_id IN
      (SELECT department_id
       FROM departments
       WHERE location_id=1700)
```

185. Seleccionar el apellido y salario de los empleados que tienen a un empleado con apellido King como jefe directo.

```
SELECT last_name, salary
FROM employees
WHERE manager_id IN
      (SELECT employee_id
       FROM employees
       WHERE last_name='King')
```

186. Seleccionar el apellido y salario de los empleados que tienen a King, (que no tiene jefe) como jefe directo.

```
SELECT last_name, salary
FROM employees
WHERE manager_id =
      (SELECT employee_id
       FROM employees
       WHERE last_name='King'
       AND manager_id IS NULL)
```

187. Seleccionar ID departamento, apellido y cod oficio de los empleados que pertenecen al departamento Executive

```
SELECT department_id, last_name, job_id
FROM employees
JOIN departments USING (department_id)
WHERE department_name = 'Executive'
```

O también:

```
SELECT department_id,
last_name,
job_id
FROM employees
WHERE department_id=
      (SELECT department_id
       FROM departments
       WHERE department_name='Executive')
```

188. Seleccionar ID empleado, apellido, sueldo de los colegas de empleados que tengan una 'u' en su apellido, y que ganen sueldos (los colegas) mayor que la media de sueldos de la empresa

```
SELECT employee_id "Nº empleado", last_name Apellido, salary Salario
FROM employees
WHERE department_id IN
      (SELECT department_id
       FROM employees
       WHERE last_name LIKE '%u%')
AND salary >
      (SELECT AVG(salary)
       FROM employees);
```

189. Seleccionar apellido y sueldo de los trabajadores pertenecientes a departamentos radicados en Seattle

```
SELECT last_name Apellido,
salary Salario
FROM employees
WHERE department_id IN
      (SELECT department_id
       FROM departments d, locations l
       WHERE d.location_id=l.location_id
       AND city='Seattle')
```

O también:

```
SELECT last_name Apellido,
salary Salario
FROM employees
WHERE department_id IN
      (SELECT department_id
       FROM departments
       WHERE location_id =
            (SELECT location_id
             FROM locations
             WHERE city='Seattle'))
```

190. Seleccionar un numero de orden y el apellido de los empleados

```
SELECT rownum,last_name
FROM employees
```

191. Seleccionar un numero y apellido de los primeros 10 empleados, sin especificar ningún orden

```
SELECT rownum,last_name
FROM employees
WHERE rownum<11
```

192. Seleccionar un numero y apellido de los primeros 10 empleados,ordenados por apellido

```
SELECT rownum,first_name
FROM employees
WHERE rownum<11
ORDER BY first_name
```

193. Seleccionar los datos de los 5 empleados con suldo más alto de la empresa

```
SELECT *
FROM
    (SELECT last_name, salary
    FROM employees
    ORDER BY salary desc)
WHERE rownum < 6
```

194. Crear un script que seleccione el apellido, sueldo y fecha contrato de los empleados que ganan más de una cantidad, que ha de proporcionar el usuario

```
SELECT last_name, salary, hire_date
FROM employees
WHERE salary > &cant;
```

195. Crear un script que calcule la media de salarios de los empleados de un departamento, que ha de proporcionar el usuario

```
SELECT AVG(salary)
FROM employees
WHERE department_id=&dpto
```

196. Crear un script que calcule la media de salarios de los empleados con un determinado cod oficio, que ha de proporcionar el usuario

```
SELECT AVG(salary)
FROM employees
WHERE job_id='&oficio'
```

197. Crear un script que calcule la media de salarios de los empleados de un departamento y cod oficios dados, que ha de proporcionar el usuario

```
SELECT AVG(salary)
FROM employees
WHERE job_id='&oficio'
AND department_id=&dpto
```

198. Crear un script que calcule la media de salarios de los empleados de un departamento y cod oficios dados, que ha de proporcionar el usuario. Posibilitar que el cod oficio se facilite en mayusculas, minusculas o como se quiera.

```
SELECT AVG(salary)
FROM employees
WHERE job_id=UPPER('&oficio')
AND department_id=&dpto;
```

199. Al ejecutar el script anterior, evitar que pregunte por oficio y dpto, y que tome como valor dpto = 80 y oficio = sa_rep. Anular estos valores después de ejecutar el script

```
DEFINE dpto = 80
DEFINE oficio = sa_rep
```

```
@script
```

```
UNDEFINE dpto
UNDEFINE oficio
```