

CAPÍTULO 12. COMPOSICIÓN DE CONSULTAS

***12.1).** Hallar el salario medio y la edad media en años de los empleados que tienen comisión y los que no.

Solución:

```
SELECT  'CON COMISION', CURRENT DATE, AVG (SALAR),
        AVG (YEAR (CURRENT DATE - FECNA) )
FROM    TEMPLE
WHERE   COMIS IS NOT NULL

UNION

SELECT  'SIN COMISION', CURRENT DATE, AVG (SALAR),
        AVG (YEAR (CURRENT DATE - FECNA) )
FROM    TEMPLE
WHERE   COMIS IS NULL
```

Resultado (suponiendo que el CURRENT DATE vale 5.7.1990):

COL-1	COL-2	COL-3	COL-4
CON COMISION	1990-07-05	2664,28	36
SIN COMISION	1990-07-05	3285,00	36

12.2). Para los empleados que no tienen comisión obtener por orden alfabético el nombre y el cociente entre su salario y el número de hijos (como el ejercicio 6.3), pero si un empleado no tiene hijos, obtener el salario sin más, indicando este caso con un literal.

Solución:

```
SELECT  NOMEM, SALAR / NUMHI, '(SALARIO / HIJOS)'
FROM    TEMPLE
WHERE   COMIS IS NULL AND NUMHI > 0

UNION

SELECT  NOMEM, SALAR, '(SALARIO)'
FROM    TEMPLE
WHERE   COMIS IS NULL AND NUMHI = 0
ORDER BY 1
```

Resultado:

NOMEM	COL-2	COL-3
AGUIRRE, AUREO	4500,00	(SALARIO)
CAMPOS, ROMULO	2000,00	(SALARIO / HIJOS)
CAMPS, AURELIO	4500,00	(SALARIO / HIJOS)
FIERRO, CLAUDIA	4000,00	(SALARIO)
FLOR, DOROTEA	580,00	(SALARIO / HIJOS)
GALVEZ, PILAR	1900,00	(SALARIO / HIJOS)
GARCIA, AUGUSTO	4200,00	(SALARIO)
GIL, GLORIA	900,00	(SALARIO / HIJOS)
LARA, LUCRECIA	1850,00	(SALARIO)
LOPEZ, ANTONIO	1200,00	(SALARIO / HIJOS)
MARTIN, MICAELA	1800,00	(SALARIO)
MORA, VALERIANA	2100,00	(SALARIO / HIJOS)
MORAN, CARMEN	2150,00	(SALARIO / HIJOS)
MUÑOZ, AZUCENA	1750,00	(SALARIO)

PEREZ, JULIO	4400,00	(SALARIO)
POLO, OTILIA	3800,00	(SALARIO)
PONS, CESAR	1033,33	(SALARIO / HIJOS)
RUIZ, FABIOLA	1900,00	(SALARIO / HIJOS)
SANZ, CORNELIO	2025,00	(SALARIO / HIJOS)
VEIGA, JULIANA	750,00	(SALARIO / HIJOS)

12.3). Para los empleados que trabajan en la calle de Atocha y su salario supera al salario medio de su departamento, obtener por orden alfabético su nombre y su salario total (salario, o salario más comisión en los que la tengan).

Solución:

```

SELECT      NOMEM, SALAR + COMIS
FROM        TEMPLE E, TDEPTO D, TCENTR C
WHERE       E.NUMDE = D.NUMDE AND D.NUMCE = C.NUMCE AND
            SEÑAS LIKE '%ATOCHA%' AND
            SALAR > (SELECT AVG (SALAR)
                     FROM TEMPLE
                     WHERE NUMDE = E.NUMDE)
            AND COMIS IS NOT NULL

UNION

SELECT      NOMEM, SALAR
FROM        TEMPLE E, TDEPTO D, TCENTR C
WHERE       E.NUMDE = D.NUMDE AND D.NUMCE = C.NUMCE AND
            SEÑAS LIKE '%ATOCHA%' AND
            SALAR > (SELECT AVG (SALAR)
                     FROM TEMPLE
                     WHERE NUMDE = E.NUMDE)
            AND COMIS IS NULL

ORDER BY    1

```

Otra formulación usando la función escalar COALESCE:

```

SELECT      NOMEM, COALESCE (SALAR + COMIS, SALAR)
FROM        TEMPLE E, TDEPTO D, TCENTR C
WHERE       E.NUMDE = D.NUMDE AND D.NUMCE = C.NUMCE AND
            SEÑAS LIKE '%ATOCHA%' AND
            SALAR > (SELECT AVG (SALAR)
                     FROM TEMPLE
                     WHERE NUMDE = E.NUMDE)

ORDER BY    NOMEM

```

Resultado:

NOMEM	COL-2
AGUIRRE, AUREO	4200
DIEZ, AMELIA	3700
GARCIA, OCTAVIO	4600
LARA, DORINDA	3500
LASA, MARIO	4600
PEREZ, MARCOS	5300
SANZ, LAVINIA	3800
TEROL, LUCIANO	4000

12.4). Hallar por departamento la masa salarial total (suma de todos los salarios y comisiones del departamento) y el nombre, por orden alfabético.

En principio se podría pensar en formularla así:

```

SELECT      NOMDE, SUM (SALAR) + SUM (COMIS)
FROM        TEMPLE E, TDEPTO D
WHERE       D.NUMDE = E.NUMDE
GROUP BY    D.NUMDE, D.NOMDE
ORDER BY    NOMDE

```

Sin embargo esta solución no es correcta para los departamentos en que ningún empleado trabaje a comisión. Para corregir este defecto se podría reformular así:

```

SELECT      NOMDE, SUM (SALAR) + SUM (COMIS)
FROM        TEMPLE E, TDEPTO D
WHERE       D.NUMDE = E.NUMDE
GROUP BY    D.NUMDE, D.NOMDE
HAVING      COUNT (DISTINCT COMIS) > 0

UNION

SELECT      NOMDE, SUM (SALAR)
FROM        TEMPLE E, TDEPTO D
WHERE       D.NUMDE = E.NUMDE
GROUP BY    D.NUMDE, D.NOMDE
HAVING      COUNT (DISTINCT COMIS) = 0

ORDER BY    1

```

Se podría también reformular con la ayuda de la función COALESCE:

```

SELECT      NOMDE,
            COALESCE (SUM (SALAR) + SUM (COMIS), SUM (SALAR))
FROM        TEMPLE E, TDEPTO D
WHERE       D.NUMDE = E.NUMDE
GROUP BY    NOMDE
ORDER BY    NOMDE

```

O también:

```

SELECT      NOMDE,
            SUM (COALESCE (SALAR + COMIS, SALAR))
FROM        TEMPLE E, TDEPTO D
WHERE       D.NUMDE = E.NUMDE
GROUP BY    NOMDE
ORDER BY    NOMDE

```

Resultado:

NOMDE	COL-2
-----	-----
DIRECCION COMERCIAL	9450
DIRECCION GENERAL	15500
FINANZAS	11100
ORGANIZACIÓN	2700
PERSONAL	12400
PROCESO DE DATOS	16200
SECTOR INDUSTRIAL	24750
SECTOR SERVICIOS	24600

12.5). Efectuar una explosión de la organización de departamentos. Es decir, para cada departamento obtener su nombre, el de los que dependen de él y el nivel al que dependen. Si un departamento depende directamente de otro este nivel será 1, si depende de uno que depende directamente de éste será 2, y así sucesivamente. Se considerará que un departamento depende de sí mismo a nivel 0. La primera columna del resultado será el nombre de un departamento, la segunda el de un departamento que depende de él, y la tercera el nivel al que depende. Considerar un máximo de 4 niveles de dependencia. Presentar el resultado por orden alfabético. Si de un departamento no depende ningún otro, aparecerá al menos dependiendo de sí mismo a nivel 0.

Solución:

```

SELECT      NOMDE, NOMDE, 0
FROM        TDEPTO

UNION

SELECT      D0.NOMDE, D1.NOMDE, 1
FROM        TDEPTO D0, TDEPTO D1
WHERE       D0.NUMDE = D1.DEPDE

UNION

SELECT      D0.NOMDE, D2.NOMDE, 2
FROM        TDEPTO D0, TDEPTO D1, TDEPTO D2
WHERE       D0.NUMDE = D1.DEPDE AND D1.NUMDE = D2.DEPDE

UNION

SELECT      D0.NOMDE, D3.NOMDE, 3
FROM        TDEPTO D0, TDEPTO D1, TDEPTO D2,
            TDEPTO D3
WHERE       D0.NUMDE = D1.DEPDE AND D1.NUMDE = D2.DEPDE AND
            D2.NUMDE = D3.DEPDE

UNION

SELECT      D0.NOMDE, D4.NOMDE, 4
FROM        TDEPTO D0, TDEPTO D1, TDEPTO D2,
            TDEPTO D3, TDEPTO D4
WHERE       D0.NUMDE = D1.DEPDE AND D1.NUMDE = D2.DEPDE AND
            D2.NUMDE = D3.DEPDE AND D3.NUMDE = D4.DEPDE

ORDER BY    1, 3, 2

```

Resultado:

NOMDE	NOMDE1	COL-3
DIRECCION COMERCIAL	DIRECCION COMERCIAL	0
DIRECCION COMERCIAL	SECTOR INDUSTRIAL	1
DIRECCION COMERCIAL	SECTOR SERVICIOS	1
DIRECCION GENERAL	DIRECCION GENERAL	0
DIRECCION GENERAL	DIRECCION COMERCIAL	1
DIRECCION GENERAL	FINANZAS	1
DIRECCION GENERAL	ORGANIZACION	1
DIRECCION GENERAL	PERSONAL	2
DIRECCION GENERAL	PROCESO DE DATOS	2
DIRECCION GENERAL	SECTOR INDUSTRIAL	2
DIRECCION GENERAL	SECTOR SERVICIOS	2
DIRECCION GENERAL	PERSONAL CONTRATADO	3
FINANZAS	FINANZAS	0
ORGANIZACION	ORGANIZACION	0
ORGANIZACION	PERSONAL	1
ORGANIZACION	PROCESO DE DATOS	1
ORGANIZACION	PERSONAL CONTRATADO	2
PERSONAL	PERSONAL	0
PERSONAL	PERSONAL CONTRATADO	1
PERSONAL CONTRATADO	PERSONAL CONTRATADO	0
PROCESO DE DATOS	PROCESO DE DATOS	0
SECTOR INDUSTRIAL	SECTOR INDUSTRIAL	0
SECTOR SERVICIOS	SECTOR SERVICIOS	0

12.6). Como el ejercicio anterior, pero añadiendo al resultado una columna con el nombre del director del departamento de la segunda columna y otra con la masa salarial total de ese departamento (salarios más comisiones).

Se puede formular usando la función escalar COALESCE o no, análogamente a como se ha hecho en ejercicios anteriores. Mostramos aquí la solución con COALESCE por ser de formulación más breve:

```

SELECT      D0.NOMDE, D0.NOMDE, 0, ED.NOMEM,
            SUM (COALESCE (EM.SALAR + EM.COMIS, EM.SALAR))
FROM        TDEPTO D0, TEMPLE ED, TEMPLE EM
WHERE       D0.DIREC = ED.NUMEM AND D0.NUMDE = EM.NUMDE
GROUP BY    D0.NUMDE, D0.NOMDE, ED.NOMEM

UNION

SELECT      D0.NOMDE, D1.NOMDE, 1, ED.NOMEM,
            SUM (COALESCE (EM.SALAR + EM.COMIS, EM.SALAR))
FROM        TDEPTO D0, TDEPTO D1,
            TEMPLE ED, TEMPLE EM
WHERE       D0.NUMDE = D1.DEPDE AND
            D1.DIREC = ED.NUMEM AND D1.NUMDE = EM.NUMDE
GROUP BY    D0.NUMDE, D0.NOMDE, D1.NUMDE, D1.NOMDE, ED.NOMEM

UNION

SELECT      D0.NOMDE, D2.NOMDE, 2, ED.NOMEM,
            SUM (COALESCE (EM.SALAR + EM.COMIS, EM.SALAR))
FROM        TDEPTO D0, TDEPTO D1, TDEPTO D2,
            TEMPLE ED, TEMPLE EM
WHERE       D0.NUMDE = D1.DEPDE AND D1.NUMDE = D2.DEPDE AND
            D2.DIREC = ED.NUMEM AND D2.NUMDE = EM.NUMDE
GROUP BY    D0.NUMDE, D0.NOMDE, D1.NUMDE, D2.NUMDE,
            D2.NOMDE, ED.NOMEM

UNION

SELECT      D0.NOMDE, D3.NOMDE, 3, ED.NOMEM,
            SUM (COALESCE (EM.SALAR + EM.COMIS, EM.SALAR))
FROM        TDEPTO D0, TDEPTO D1, TDEPTO D2, TDEPTO D3,
            TEMPLE ED, TEMPLE EM
WHERE       D0.NUMDE = D1.DEPDE AND D1.NUMDE = D2.DEPDE AND
            D2.NUMDE = D3.DEPDE AND D3.DIREC = ED.NUMEM AND
            D3.NUMDE = EM.NUMDE
GROUP BY    D0.NUMDE, D0.NOMDE, D1.NUMDE, D2.NUMDE, D3.NUMDE,
            D3.NOMDE, ED.NOMEM

UNION

SELECT      D0.NOMDE, D4.NOMDE, 4, ED.NOMEM,
            SUM (COALESCE (EM.SALAR + EM.COMIS, EM.SALAR))
FROM        TDEPTO D0, TDEPTO D1, TDEPTO D2, TDEPTO D3, TDEPTO D4,
            TEMPLE ED, TEMPLE EM
WHERE       D0.NUMDE = D1.DEPDE AND D1.NUMDE = D2.DEPDE AND
            D2.NUMDE = D3.DEPDE AND D3.NUMDE = D4.DEPDE AND
            D4.DIREC = ED.NUMEM AND D4.NUMDE = EM.NUMDE
GROUP BY    D0.NUMDE, D0.NOMDE, D1.NUMDE, D2.NUMDE, D3.NUMDE,
            D4.NUMDE, D4.NOMDE, ED.NOMEM
ORDER BY    1, 3, 2

```

Resultado:

NOMDE	NOMDE1	COL-3	NOMEM	COL-5
DIRECCION COMERCIAL	DIRECCION COMERCIAL	0	PEREZ, MARCOS	9450
DIRECCION COMERCIAL	SECTOR INDUSTRIAL	1	PEREZ, MARCOS	24750
DIRECCION COMERCIAL	SECTOR SERVICIOS	1	GARCIA, OCTAVIO	24600
DIRECCION GENERAL	DIRECCION GENERAL	0	LOPEZ, ANTONIO	15500
DIRECCION GENERAL	DIRECCION COMERCIAL	1	PEREZ, MARCOS	9450
DIRECCION GENERAL	FINANZAS	1	GARCIA, AUGUSTO	11100
DIRECCION GENERAL	ORGANIZACION	1	PEREZ, JULIO	2700
DIRECCION GENERAL	PERSONAL	2	PEREZ, JULIO	12400
DIRECCION GENERAL	PROCESO DE DATOS	2	CAMPS, AURELIO	16200
DIRECCION GENERAL	SECTOR INDUSTRIAL	2	PEREZ, MARCOS	24750
DIRECCION GENERAL	SECTOR SERVICIOS	2	GARCIA, OCTAVIO	24600
FINANZAS	FINANZAS	0	GARCIA, AUGUSTO	11100
ORGANIZACION	ORGANIZACION	0	PEREZ, JULIO	2700
ORGANIZACION	PERSONAL	1	PEREZ, JULIO	12400
ORGANIZACION	PROCESO DE DATOS	1	CAMPS, AURELIO	16200
PERSONAL	PERSONAL	0	PEREZ, JULIO	12400
PROCESO DE DATOS	PROCESO DE DATOS	0	CAMPS, AURELIO	16200
SECTOR INDUSTRIAL	SECTOR INDUSTRIAL	0	PEREZ, MARCOS	24750
SECTOR SERVICIOS	SECTOR SERVICIOS	0	GARCIA, OCTAVIO	24600

Al aparecer la tabla de Empleados en la yunción, desaparecen 4 filas respecto al ejercicio anterior; corresponden a aquéllas en las que figura el departamento de PERSONAL CONTRATADO, y que no tiene empleados.

12.7). Supongamos que algunos departamentos se van a trasladar a otro local. Disponemos de una tabla llamada TTRASL con una sola columna llamada NUMDE donde hay una fila por cada departamento que se traslada al local nuevo. Se desea producir una lista por orden alfabético de todos los departamentos, indicando cuáles se trasladan y cuáles no. Este tipo de operaciones en que se combinan filas de una tabla con las de otra, incluyendo en el resultado también las que no están emparejadas, se conoce en teoría de Bases de Datos relacionales como yunción externa ("outer join"), que se tratará en un capítulo posterior, pero aquí pedimos formularla utilizando el predicado EXISTS y la cláusula UNION. En la tabla TTRASL hemos dado de alta los departamentos 100 y 110.

Solución:

```

SELECT D.*, 'SE TRASLADA'
FROM   TDEPTO D, TTRASL T
WHERE  D.NUMDE = T.NUMDE

UNION

SELECT D.*, 'NO SE TRASLADA'
FROM   TDEPTO D
WHERE  NOT EXISTS (SELECT *
                   FROM   TTRASL T
                   WHERE  D.NUMDE = T.NUMDE)

ORDER BY 7

```

Resultado:

NUMDE	NUMCE	DIREC	TIDIR	PRESU	DEPDE	NOMDE	COL-8
110	20	180	P	150	100	DIRECCION COMERCIAL	SE TRASLADA
100	10	260	P	120	-	DIRECCION GENERAL	SE TRASLADA
130	10	310	P	20	100	FINANZAS	NO SE TRASLADA
120	10	150	F	30	100	ORGANIZACION	NO SE TRASLADA
121	10	150	P	20	120	PERSONAL	NO SE TRASLADA
123	-	150	F	100	121	PERSONAL CONTRATADO	NO SE TRASLADA
122	10	350	P	60	120	PROCESO DE DATOS	NO SE TRASLADA
111	20	180	F	110	110	SECTOR INDUSTRIAL	NO SE TRASLADA
112	20	270	P	90	110	SECTOR SERVICIOS	NO SE TRASLADA

CAPÍTULO 13. EXPRESIONES DE TABLA ANIDADA

13.1). Sin hacer uso de la cláusula de agrupamiento de filas se desea obtener una lista de los empleados del departamento 121, su salario y el mayor salario existente dentro del conjunto completo de los empleados.

Solución:

```
SELECT NUMEM, SALAR,  
      (SELECT MAX(SALAR) FROM TEMPLE) AS MAXSALAR  
FROM TEMPLE  
WHERE NUMDE = 121
```

ó

```
SELECT NUMEM, SALAR, MAXSALAR  
FROM TEMPLE,  
      (SELECT MAX(SALAR) AS MAXSALAR FROM TEMPLE) AS XXX  
WHERE NUMDE = 121
```

Resultado:

NUMEM	SALAR	MAXSALAR
-----	-----	-----
110	3100	7200
150	4400	7200
190	3000	7200
370	1900	7200

13.2). Lo mismo que el anterior pero especificando el mayor salario del departamento.

Solución:

```
SELECT NUMEM, SALAR,  
      (SELECT MAX(B.SALAR) FROM TEMPLE B  
       WHERE B.NUMDE=A.NUMDE) AS MAXSALAR  
FROM TEMPLE A  
WHERE NUMDE = 121;
```

ó

```
SELECT NUMEM, SALAR, MAXSALAR  
FROM TEMPLE A, TABLE (SELECT MAX(B.SALAR) AS MAXSALAR  
                       FROM TEMPLE B  
                       WHERE B.NUMDE=A.NUMDE) AS XXX  
WHERE NUMDE = 121;
```

Resultado:

NUMEM	SALAR	MAXSALAR
-----	-----	-----
110	3100	4400
150	4400	4400
190	3000	4400
370	1900	4400

13.3). Hallar los departamentos cuyo presupuesto supera al presupuesto medio de los departamentos que dependen del mismo del que depende él, incluido él mismo.

```

SELECT D1.NUMDE, D1.PRESU, PREMED
FROM TDEPTO D1,
     TABLE (SELECT AVG (PRESU) AS PREMED
              FROM TDEPTO D2
              WHERE D2.DEPDE = D1.DEPDE
            ) AS D3
WHERE D1.PRESU > PREMED
ORDER BY D1.NUMDE

```

Resultado:

NUMDE	PRESU	PREMED
110	150	66,66
111	110	100,00
122	60	40,00

Formulación equivalente:

```

SELECT D1.NUMDE, D1.PRESU, AVG (D2.PRESU) AS PREMED
FROM TDEPTO D1, TDEPTO D2
WHERE D1.DEPDE = D2.DEPDE
GROUP BY D1.NUMDE, D1.PRESU
HAVING D1.PRESU > AVG (D2.PRESU)
ORDER BY D1.NUMDE

```

CAPÍTULO 14. EXPRESIONES CONDICIONALES (“CASE”)

14.1). Obtener con el uso de expresiones CASE el número de empleados nacidos antes del 67 y los que ingresaron en la compañía después del 84.

Solución:

```

SELECT SUM (CASE WHEN FECNA < '01.01.1967' THEN 1 ELSE 0 END),
       SUM (CASE WHEN FECIN > '31.12.1984' THEN 1 ELSE 0 END)
FROM TEMPLE

```

Resultado:

COL-1	COL-2
28	14

14.2). Obtener cuántos empleados con fecha de nacimiento superior al 1 de enero de 1980; incluir en la consulta la media de los salarios. Buscar dos columnas adicionales (usando funciones o expresiones conocidas) a incluir en la consulta que permitan transformar los resultados nulos de la media de salarios en valor cero.

Solución:

```

SELECT COUNT (*) AS C1, AVG (SALAR) AS A1,
       COALESCE ( AVG (SALAR), 0 ) AS A2,
       CASE WHEN AVG (SALAR) IS NULL THEN 0 ELSE AVG (SALAR)
       END AS A3
FROM TEMPLE
WHERE FECNA > '1.1.1980'

```

Resultado:

C1	A1	A2	A3
----	----	----	----

0 - 0 0

14.3). Obtener por grupos de empleados, los de los números de empleado del 100 al 199, los del 200 al 299, y el resto, los máximos salarios, los mínimos y las medias.

Solución:

```
SELECT CASE_NUMEM, MAX (SALAR) AS MAXS, MIN (SALAR) AS MINS,
       AVG (SALAR) AS AVGS
FROM   (SELECT SALAR,
       CASE WHEN NUMEM BETWEEN 100 AND 199 THEN 'LOS 100S'
       WHEN NUMEM BETWEEN 200 AND 299 THEN 'LOS 200S'
       END AS CASE_NUMEM
       FROM TEMPLE) AS X
GROUP BY CASE_NUMEM
```

Resultado:

CASE_NUMEM	MAXS	MINS	AVGS
LOS 100S	4800	2900	3542,85
LOS 200S	7200	2700	3937,50
-	4500	1000	2457,89

14.4). Obtener los números de departamento de aquellos con presupuestos superiores a 90.000 € mostrando el presupuesto inicial y el nuevo teniendo en cuenta que se incrementa en un 10% para los que tienen dependencia de DIRECCIÓN GENERAL, y el 5% al resto. Además se sube un 3% adicional si tiene director EN PROPIEDAD.

Solución:

```
SELECT NUMDE, PRESU,
       PRESU + PRESU*((CASE WHEN DEPDE = 100 THEN 0.10
       ELSE 0.05 END) +
       (CASE WHEN TIDIR = 'P' THEN 0.03
       ELSE 0.00 END)
       )
       AS NUEVOPRESU
FROM TDEPTO
WHERE PRESU > 90
```

Resultado:

NUMDE	PRESU	NUEVOPRESU
100	120	129,60
110	150	169,50
111	110	115,50
123 100		105,00

14.5). Transformar la sentencia SQL anterior en otra que haga uso de la cláusula UNION.

Solución:

```
SELECT NUMDE, PRESU, PRESU*1.13 AS NUEVOPRESU
FROM TDEPTO
WHERE DEPDE = 100 AND TIDIR = 'P' AND PRESU > 90
UNION ALL
SELECT NUMDE, PRESU, PRESU*1.10 AS NUEVOPRESU
FROM TDEPTO
WHERE DEPDE = 100 AND TIDIR = 'F' AND PRESU > 90
UNION ALL
SELECT NUMDE, PRESU, PRESU*1.08 AS NUEVOPRESU
```

```

FROM TDEPTO
WHERE (DEPDE <> 100 OR DEPDE IS NULL) AND TIDIR = 'P' AND PRESU > 90
UNION ALL
SELECT NUMDE, PRESU, PRESU*1.05 AS NUEVOPRESU
FROM TDEPTO
WHERE (DEPDE <> 100 OR DEPDE IS NULL) AND TIDIR = 'F' AND PRESU > 90

```

14.6). Obtener los números de empleado, sus departamentos y la suma de salarios y comisiones, para aquellos cuyo salario sea mayor de 4 veces lo que ingresa por comisiones y que no estén pagados sólo en salario.

```

SELECT NUMEM, NUMDE, (SALAR + COMIS) AS "SALARIO+COMISIONES"
FROM TEMPLE
WHERE (CASE WHEN COMIS IS NULL THEN NULL
            WHEN COMIS = 0 THEN NULL
            ELSE SALAR / COMIS
        END) > 4
ORDER BY NUMEM

```

Resultado:

NUMEM	NUMDE	SALARIO+COMISIONES
180	110	5300
270	112	4600

14.7). Formar tres grupos con los departamentos. En uno se incluirán los departamentos 110, 120 y 130, y lo llamaremos “GRUPO A”. En otro los departamentos 111, 112, 121 y 122, y será el “GRUPO B”. El resto se incluye en un grupo llamado “OTROS”. Para cada grupo hallar el presupuesto máximo, el mínimo y el medio.

```

SELECT GRUPOS,
       MAX (PRESU) AS MAX, MIN (PRESU) AS MIN, AVG (PRESU) AS MEDIA
FROM (SELECT PRESU, CASE
        WHEN NUMDE IN (110,120,130) THEN 'GRUPO A'
        WHEN NUMDE IN (111,112,121,122) THEN 'GRUPO B'
        ELSE 'OTROS'
      END AS GRUPOS
      FROM TDEPTO) AS X
GROUP BY GRUPOS
ORDER BY GRUPOS

```

Resultado:

GRUPOS	MAX	MIN	MEDIA
GRUPO A	150	20	66,66
GRUPO B	110	20	70,00
OTROS	120	120	110,00

CAPÍTULO 15. OTRA MANERA DE ESPECIFICAR LAS YUNCIONES (“JOINS”)

15.1). Obtener la lista de los departamentos que no tienen empleados asignados. Mostrar el número y nombre.

Solución:

```

SELECT D.NUMDE AS DEPTO, NOMDE
FROM TDEPTO D LEFT OUTER JOIN TEMPLE E
ON D.NUMDE = E.NUMDE
EXCEPT
SELECT D.NUMDE AS DEPTO, NOMDE

```

```

FROM TDEPTO D INNER JOIN TEMPLE E
      ON D.NUMDE = E.NUMDE
ORDER BY DEPTO

```

Resultado:

DEPTO	NOMDE
-----	-----
123	PERSONAL CONTRATADO

O también:

```

SELECT NUMDE AS DEPTO, NOMDE
FROM TDEPTO
WHERE NOT EXISTS ( SELECT
                    FROM TEMPLE
                    WHERE TEMPLE.NUMDE = TDEPTO.NUMDE)
ORDER BY DEPTO

```

15.2). Hallar cuántos empleados hay que no tengan departamento asignado.

```

SELECT COUNT (*) AS EMPLES
FROM (SELECT NUMEM
      FROM TEMPLE E LEFT OUTER JOIN TDEPTO D
      ON E.NUMDE = D.NUMDE
      EXCEPT
      SELECT NUMEM
      FROM TEMPLE E INNER JOIN TDEPTO D
      ON E.NUMDE = D.NUMDE
      ) AS TD

```

Resultado :

EMPLES

0

O también :

```

SELECT COUNT (*) AS EMPLES
FROM (SELECT NUMEM
      FROM TEMPLE E
      WHERE NOT EXISTS ( SELECT NUMDE
                        FROM TDEPTO D
                        WHERE D.NUMDE = E.NUMDE)
      ) AS T

```

15.3). Para cada centro, obtener la suma de los salarios de los empleados que trabajan en él. Los centros que no tengan empleados se mostrarán con una suma cero.

```

SELECT C.NUMCE AS CENTRO, COALESCE (SUM (SALAR), 0) AS SUMA
FROM (TCENTR C LEFT OUTER JOIN TDEPTO D
      ON C.NUMCE = D.NUMCE )
      LEFT OUTER JOIN TEMPLE E
      ON D.NUMDE = E.NUMDE
GROUP BY C.NUMCE
ORDER BY CENTRO

```

Resultado :

CENTRO	SUMA
-----	-----

10	57900
20	45100
50	0

15.4). Hallar los departamentos que tienen un presupuesto mayor que el del departamento del cual dependen. Incluir también los departamentos que no dependan de otro. Mostrar número de departamento, número de departamento del que dependen y presupuestos de ambos.

```
SELECT D1.NUMDE AS DEPTO, D1.DEPDE AS DEPEND, D1.PRESU AS PRESU1,
D2.PRESU AS PRESU2
FROM TDEPTO D1 LEFT OUTER JOIN TDEPTO D2
ON D1.DEPDE = D2.NUMDE
WHERE (D1.PRESU>D2.PRESU OR D2.PRESU IS NULL)
ORDER BY DEPTO
```

Resultado :

DEPTO	DEPEND	PRESU1	PRESU2
-----	-----	-----	-----
100	-	120	-
110	100	150	120
122	120	60	30
123	121	100	120

Con la yunción externa se obtienen todos los emparejamientos de dependencias incluida la del departamento 100 que no depende de nadie. Mediante la cláusula WHERE se filtran, por un lado aquellas dependencias donde el presupuesto del departamento primero es mayor del del departamento del que dependen, y por otro mantenemos el departamento 100 porque el presupuesto del departamento del que depende (ninguno) es nulo.

15.5). Para los empleados de los departamentos 100 y 110, obtener un listado de ellos que incluya: nombre y salario medio de los que cobran más que él, sea cual sea su departamento. Mostrar también los que no tengan a nadie con mayor salario que el suyo.

```
SELECT T1.NOMEM AS NOMBRE, AVG (E2.SALAR) AS SAL_MEDIO
FROM (SELECT * FROM TEMPLE WHERE NUMDE IN (100,110)) AS T1
LEFT OUTER JOIN TEMPLE E2
ON T1.SALAR < E2.SALAR
GROUP BY T1.NOMEM
ORDER BY NOMBRE
```

Resultado :

NOMEM	SAL_MEDIO
-----	-----
ALBA, ADRIANA	6000,00
CAMPOS, ROMULO	3419,23
GALVEZ, PILAR	4706,25
LOPEZ, ANTONIO	-
MORAN, CARMEN	3730,95
PEREZ, MARCOS	7200,00

15.6). Considerando solo los empleados de los departamentos 100 y 110, obtener un listado de ellos que incluya: nombre y salario medio de los que cobran más que él. Mostrar también los que no tengan a nadie con mayor salario que el suyo.

```
SELECT T1.NOMEM AS NOMBRE, AVG (T2.SALAR) AS SAL_MEDIO
FROM (SELECT * FROM TEMPLE WHERE NUMDE IN (100,110)) AS T1
LEFT OUTER JOIN
(SELECT * FROM TEMPLE WHERE NUMDE IN (100,110)) AS T2
ON T1.SALAR < T2.SALAR
GROUP BY T1.NOMEM
ORDER BY NOMBRE
```

Resultado :

NOMEM	SAL_MEDIO
-----	-----
ALBA, ADRIANA	6000,00
CAMPOS, ROMULO	4490,00
GALVEZ, PILAR	5500,00
LOPEZ, ANTONIO	-
MORAN, CARMEN	5075,00
PEREZ, MARCOS	7200,00

15.7). Considerando solo los empleados de los departamentos 100 y 110, obtener un listado de ellos que incluya: nombre, comisión, y cuántos empleados de estos departamentos tienen una comisión mayor que la suya. Este valor se mostrará como un *Nulo* en los que no tengan comisión.

```

SELECT T1.NOMEM AS NOMBRE, T1.COMIS AS COMISION,
      CASE
        WHEN T1.COMIS IS NOT NULL AND MAX (T2.COMIS) IS NULL
        THEN 0
        WHEN T1.COMIS IS NOT NULL THEN COUNT (*)
        ELSE T1.COMIS
      END AS EMPLES
FROM (SELECT * FROM TEMPLE WHERE NUMDE IN (100,110)) AS T1
LEFT OUTER JOIN
      (SELECT * FROM TEMPLE WHERE NUMDE IN (100,110)) AS T2
ON T1.COMIS < T2.COMIS
GROUP BY T1.NOMEM, T1.COMIS
ORDER BY NOMBRE

```

Resultado :

NOMBRE	COMISION	EMPLES
-----	-----	-----
ALBA, ADRIANA	-	-
CAMPOS, ROMULO	-	-
GALVEZ, PILAR	-	-
LOPEZ, ANTONIO	-	-
MORAN, CARMEN	-	-
PEREZ, MARCOS	500	0

15.8). Contestar al ejercicio 12.7 empleando la yunción externa. Recordemos su enunciado: Supongamos que algunos departamentos se van a trasladar a otro local. Disponemos de una tabla llamada TTRASL con una sola columna llamada NUMDE donde hay una fila por cada departamento que se traslada al local nuevo. Se desea producir una lista por orden alfabético de todos los departamentos, indicando cuáles se trasladan y cuáles no. En la tabla TTRASL hemos dado de alta los departamentos 100 y 110.

```

SELECT D.NOMDE,
      CASE
        WHEN T.NUMDE IS NULL THEN 'NO SE TRASLADA'
        ELSE 'SE TRASLADA'
      END AS TRASLADO
FROM TDEPTO D LEFT OUTER JOIN TTRASL T
ON D.NUMDE = T.NUMDE
ORDER BY NOMDE

```

Resultado :

NOMDE	TRASLADO
-----	-----

DIRECCION COMERCIAL	SE TRASLADA
DIRECCION GENERAL	SE TRASLADA
FINANZAS	NO SE TRASLADA
ORGANIZACION	NO SE TRASLADA
PERSONAL	NO SE TRASLADA
PERSONAL CONTRATADO	NO SE TRASLADA
PROCESO DE DATOS	NO SE TRASLADA
SECTOR INDUSTRIAL	NO SE TRASLADA
SECTOR SERVICIOS	NO SE TRASLADA

15.9). Obtener el conjunto de empleados con su número, departamento al que pertenecen y director del departamento aunque no exista el número del departamento. Finalmente se seleccionarán sólo las filas donde el departamento del empleado sea el 110, y donde los empleados tengan un salario inferior a los 3.000 euros.

Solución:

```
SELECT NUMEM, A.NUMDE, DIREC
FROM TEMPLE A LEFT OUTER JOIN TDEPTO B
ON A.NUMDE=B.NUMDE
WHERE A.NUMDE = 110 AND SALAR < 3000
```

Resultado:

NUMEM	NUMDE	DIREC
-----	-----	-----
390	110	180
510	110	180

CAPÍTULO 16. EXPRESIONES DE TABLA COMÚN Y RECURSIVIDAD

16.1). Obtener el valor mínimo de las medias de salario de los empleados por departamento.

Solución:

```
WITH TEMP1 AS
  (SELECT NUMDE,
   AVG (SALAR) AS AVGSALAR
   FROM TEMPLE
   GROUP BY NUMDE),
TEMP2 AS
  (SELECT MIN (AVGSALAR) AS MINSALAR
   FROM TEMP1)

SELECT * FROM TEMP2
```

Resultado:

MINSALAR

2181.25

16.2). Obtener el conjunto de departamentos cuyos empleados tengan una media de número de hijos inferior a la media de todos los empleados.

Solución:

```
WITH TEMP1 AS
  (SELECT AVG (NUMHI) AS MEDIAHID, NUMDE AS #NUMDE
   FROM TEMPLE
   GROUP BY NUMDE),
```

```
TEMP2 (MEDIAHI) AS
      (SELECT AVG (NUMHI)
       FROM TEMPLE)
```

```
SELECT NUMDE, NOMDE FROM TDEPTO, TEMP1, TEMP2
WHERE NUMDE = #NUMDE AND MEDIAHID < MEDIAHI
```

Resultado:

NUMDE	NOMDE
112	SECTOR SERVICIOS
122	PROCESO DE DATOS

16.3). Reformular la sentencia SQL anterior haciendo uso de la subselección en cláusula FROM.

Solución:

```
SELECT NUMDE, NOMDE
FROM TDEPTO, (SELECT AVG (NUMHI) AS MEDIAHID, NUMDE AS #NUMDE
              FROM TEMPLE
              GROUP BY NUMDE) AS T1,
              (SELECT AVG (NUMHI) AS MEDIAHI FROM TEMPLE) AS T2
WHERE NUMDE = #NUMDE AND MEDIAHID < MEDIAHI
```

16.4). Obtener para un número de hijos entre 0 y 9, cuántos empleados hay.

Solución:

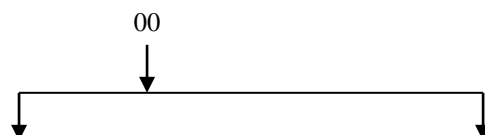
```
WITH NUM_HIJOS (NHIJOS) AS
      (VALUES (0),(1),(2),(3),(4),(5), (6), (7), (8), (9))

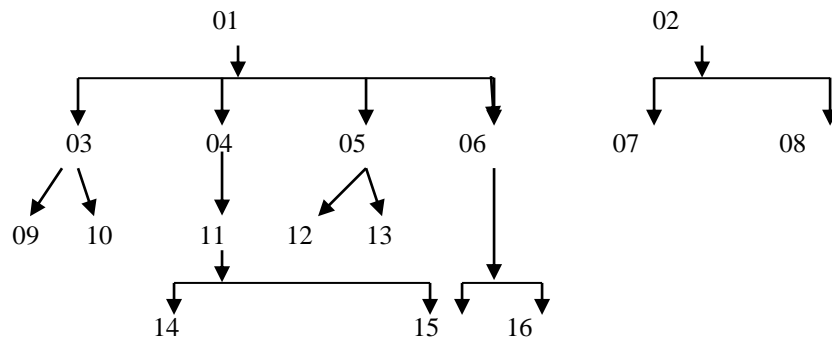
SELECT NHIJOS, COALESCE (NUMEMPS,0) AS NUMEMPS
FROM NUM_HIJOS LEFT OUTER JOIN (SELECT NUMHI, COUNT(*) AS NUMEMPS
                                FROM TEMPLE
                                WHERE NUMHI < 10
                                GROUP BY NUMHI) AS XXX
ON NHIJOS = NUMHI
ORDER BY 1
```

Resultado:

NHIJOS	NUMEMPS
0	14
1	7
2	6
3	4
4	1
5	1
6	1
7	0
8	0
9	0

16.5). Mostrar una tabla con el conjunto de filas que reflejen los elementos de la jerarquía del siguiente gráfico. Se trata de un diagrama de piezas y subpiezas que las forman.





Solución:

PIEZA	SUBPIEZA
00	01
00	02
01	03
01	04
01	05
01	06
02	07
02	08
03	09
03	10
04	11
05	12
05	13
11	14
11	15
06	15
06	16

16.6). Obtener el listado de las piezas dependientes de la pieza 04.

Solución:

```

WITH TABLA1 (PIEZA, SUBPIEZA) AS
  (SELECT PIEZA, SUBPIEZA
   FROM PIEZAS
   WHERE PIEZA = '04'

   UNION ALL

   SELECT C.PIEZA, C.SUBPIEZA
   FROM PIEZAS C, TABLA1 P
   WHERE P.SUBPIEZA = C.PIEZA
  )

SELECT PIEZA, SUBPIEZA
FROM TABLA1
  
```

Resultado

PIEZA	SUBPIEZA
04	11
11	14
11	15

16.7). ¿ Cuántas piezas distintas hay dependientes de la 00?.

Solución:

```
WITH TABLA1 (PIEZA, SUBPIEZA) AS
  (SELECT PIEZA, SUBPIEZA
   FROM PIEZAS
   WHERE PIEZA = '00'

   UNION ALL

   SELECT C.PIEZA, C.SUBPIEZA
   FROM PIEZAS C, TABLA1 P
   WHERE P.SUBPIEZA = C.PIEZA
  )

SELECT COUNT (DISTINCT SUBPIEZA)
FROM PIEZAS
```

Resultado:

```
COL-1
-----
    16
```

16.8). ¿ Qué subpieza forma parte de más de una pieza?.

Solución:

```
WITH TABLA2 (SUBPIEZA) AS
  (SELECT SUBPIEZA
   FROM PIEZAS
   WHERE PIEZA = '00'

   UNION ALL

   SELECT C.SUBPIEZA
   FROM PIEZAS C, TABLA2 P
   WHERE P.SUBPIEZA = C.PIEZA
  )

SELECT SUBPIEZA
FROM TABLA2
GROUP BY SUBPIEZA
HAVING COUNT(*) > 1
```

Resultado:

```
SUBPIEZA
-----
    15
```

16.9). ¿ Cuáles son las subpiezas de ·último nivel?.

Solución:

```
WITH TABLA3 (SUBPIEZA, NIVEL) AS
  (SELECT SUBPIEZA, 1
   FROM PIEZAS
   WHERE PIEZA = '00'

   UNION ALL
```

```

SELECT C.SUBPIEZA, P.NIVEL+1
FROM PIEZAS C, TABLA3 P
WHERE P.SUBPIEZA = C.PIEZA
)

```

```

SELECT SUBPIEZA, NIVEL
FROM TABLA3
WHERE NIVEL = (SELECT MAX (NIVEL) FROM TABLA3)

```

Resultado:

SUBPIEZA	NIVEL
-----	-----
14	4
15	4

16.10). Mostrar las piezas de nivel 2 con las piezas de las que dependen.

Solución:

```

WITH TABLA4 (PIEZA, SUBPIEZA, NIVEL) AS
  (SELECT PIEZA, SUBPIEZA, 1
   FROM PIEZAS
   WHERE PIEZA = '00'

   UNION ALL

   SELECT C.PIEZA, C.SUBPIEZA, P.NIVEL+1
   FROM PIEZAS C, TABLA4 P
   WHERE P.SUBPIEZA = C.PIEZA
        AND P.NIVEL+1 < 3
  )

SELECT DISTINCT SUBPIEZA, PIEZA
FROM TABLA4
WHERE NIVEL = 2

```

Resultado:

SUBPIEZA	PIEZA
-----	-----
03	01
04	01
05	01
06	01
07	02
08	02

16.11). Supongamos que hay dos tablas, llamadas TCURSO y TCURRI. La primera tiene una sola columna llamada NOMCU y la segunda tiene dos, llamadas NOMEM y NOMCU. La tabla TCURSO tiene una fila por cada curso de idiomas impartido a los empleados de la empresa, con el nombre del curso correspondiente en la columna NOMCU. La TCURRI tiene una fila por cada empleado y curso, con el nombre del empleado en la columna NOMEM y el de curso en la NOMCU. El significado de una fila de TCURRI (cuyo nombre viene de curriculum) es que el empleado ha asistido al curso correspondiente.

Se desea hallar una relación de los nombres de los empleados que han asistido a todos los cursos impartidos. En el ej. 11.32 se resolvió esta consulta mediante sentencias correlacionadas. Resolverla ahora mediante expresiones de tabla común

Este tipo de consultas en las que se buscan los valores de una tabla que están combinados con todos los de otra se llama DIVISION en Teoría de Bases de Datos Relacionales.

Solución:

```

WITH T1 (NOMEM) AS
    (SELECT DISTINCT NOMEM
     FROM TCURRI),
T2 (NOMEM, NOMCU) AS
    (SELECT * FROM T1, TCURSO
     EXCEPT
     SELECT * FROM TCURRI)

SELECT * FROM T1
EXCEPT
SELECT DISTINCT NOMEM FROM T2

```

Supongamos los siguientes contenidos de las tablas (cursos de inglés, I, y francés, F):

TCURSO)	NOMCU	TCURRI)	NOMEM	NOMCU
	-----		-----	-----
	I		PEPE	I
	F		PACO	I
			PACO	F

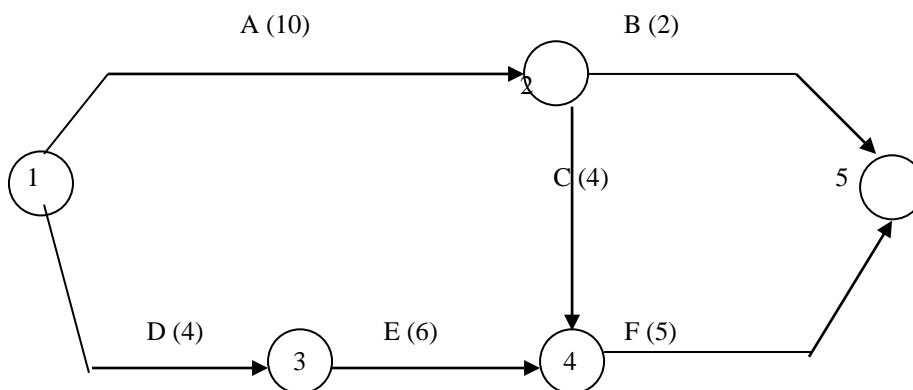
Resultado:

```

NOMEM
-----
PACO

```

16.12) El siguiente diagrama PERT (Project Evaluation and Review Technique) representa las actividades de un proyecto.



Cada arco del grafo representa una actividad, con su nombre y duración entre paréntesis. Los nodos representan precedencia entre las actividades. Así por ejemplo la actividad F, que sale del nodo 4, no puede empezar hasta que hayan terminado completamente las que llegan al nodo, es decir, la C y la E.

Este grafo se almacena en una tabla TPROY (ORG, DES, NOM, DUR). Cada fila es un arco, con ORG = nodo origen, DEL = nodo destino, NOM = nombre, DUR = duración. Contenido:

ORG	DES	NOM	DUR
-----	-----	-----	-----
1	2	A	10
1	3	D	4
2	5	B	2
2	4	C	4
3	4	E	6
4	5	F	5

Se desea hallar la duración del proyecto.

Solución:

La duración del proyecto es la del camino más largo del grafo yendo desde el nodo inicial al final.

```
WITH CAMINOS (DES, TRAMOS, DUR) AS
  (SELECT DES, 1, DUR
   FROM TPROY
   WHERE ORG = 1
   UNION ALL
   SELECT P.DES, TRAMOS + 1, C.DUR + P.DUR
   FROM CAMINOS C INNER JOIN TPROY P
    ON C.DES = P.ORG AND AND TRAMOS < 10)
SELECT MAX (DUR) AS DURACION
FROM CAMINOS
WHERE DES = 5
```

Aunque un diagrama de este tipo no puede tener ciclos, hemos incluido un control de 10 tramos como máximo para evitar este caso si se presenta por error.

Resultado:

```
DURACION
-----
19
```