

Solución Elmasri & Navathe 8.13 (7.18)

- a. Empleados del departamento 5 que trabajan más de 10 horas/sem. en el proyecto 'ProductoX'.

Con la sintaxis 1 para FROM (indicando las condiciones de reunión en WHERE)

```
SELECT NOMBRE, APELLIDO  
FROM EMPLEADO, PROYECTO, TRABAJA_EN  
WHERE NSS=NSSE AND NUMEROP=NP AND  
      ND=5 AND NOMBREP='ProductoX'  
AND HORAS>10;
```

En el examen de SQL habrá
que usar
OBLIGATORIAMENTE
esta sintaxis 2

Con la sintaxis 2 para FROM (indicando los tipos de reunión-JOIN)

```
SELECT NOMBRE, APELLIDO  
FROM (EMPLEADO INNER JOIN TRABAJA_EN  
      ON NSS=NSSE ) INNER JOIN PROYECTO  
      ON NP=NUMEROP  
WHERE ND=5 AND NOMBREP='ProductoX'  
AND HORAS>10;
```

Solución Elmasri &Navathe 8.13 (7.18)

- b. Empleados con un dependiente con igual nombre de pila.

Para que nos salga alguna solución insertamos una nueva fila:

```
INSERT INTO DEPENDIENTE(NSSE,NOMBRE_DEPENDIENTE) VALUES  
('123456789','John');
```

Con la sintaxis 1 para FROM

```
SELECT NOMBRE, APELLIDO  
FROM EMPLEADO, DEPENDIENTE  
WHERE NSS=NSSE AND  
NOMBRE=NOMBRE_DEPENDIENTE;
```

Con la sintaxis 2 para FROM

```
SELECT NOMBRE, APELLIDO  
FROM EMPLEADO INNER JOIN DEPENDIENTE  
ON NSS=NSSE AND NOMBRE=NOMBRE_DEPENDIENTE;
```

Solución Elmasri & Navathe 8.13 (7.18) (2)

- c. Empleados cuyo jefe directo es Franklin Wong.

Con la sintaxis 1 para FROM

```
SELECT E.NOMBRE, E.APELLIDO  
FROM EMPLEADO AS E, EMPLEADO AS JEFE  
WHERE JEFE.NSS=E.NSS_SUPERV AND  
      JEFE.NOMBRE='Franklin' AND  
      JEFE.APELLIDO='Wong' ;
```

Con la sintaxis 2 para FROM

```
SELECT E.NOMBRE, E.APELLIDO  
FROM EMPLEADO AS E INNER JOIN EMPLEADO AS JEFE  
      ON E.NSS_SUPERV =JEFE.NSS  
WHERE JEFE.NOMBRE='Franklin' AND  
      JEFE.APELLIDO='Wong' ;
```

Solución Elmasri &Navathe 8.13 (7.18) (2)

- d. Nombre de cada proyecto junto al número total de horas trabajadas por los empleados en él.

```
SELECT NOMBREP, SUM(HORAS) AS TOTAL_HORAS  
FROM PROYECTO INNER JOIN TRABAJA_EN ON NUMEROP=NP  
GROUP BY NOMBREP ;
```

Obtener solamente los que trabajan más de 40 horas. Añadir al final de lo anterior

```
HAVING SUM(HORAS)>40
```

- e. Apellidos y nombres de todos los empleados junto con el nombre del proyecto en el que trabajan. Resultados ordenados por apellido y proyecto.

```
SELECT APELLIDO, NOMBRE, NOMBREP  
FROM (EMPLEADO INNER JOIN TRABAJA_EN ON NSS=NSSE) INNER JOIN  
    PROYECTO ON NP=NUMEROP  
ORDER BY APELLIDO, NOMBREP;
```

Solución Elmasri & Navathe 8.13 (7.18)

- f. Empleados que no trabajan en ningún proyecto.

Para que nos salga alguna solución insertamos una nueva fila:

```
INSERT INTO EMPLEADO(NSS,APELLIDO,NOMBRE) VALUES('0','NADENA','NOKURRO');
```

```
SELECT NOMBRE,APELLIDO
```

```
FROM EMPLEADO
```

```
WHERE NSS NOT IN (SELECT NSSE FROM TRABAJA_EN);
```

Una vez probado se puede borrar la fila con:

```
DELETE FROM EMPLEADO WHERE NSS='0';
```

Solución Elmasri &Navathe 8.13 (7.18) (3)

- g.** Nombre de cada departamento junto al salario medio de los empleados asignados al mismo.

```
SELECT NOMBRED, AVG(SALARIO) AS SALARIO_MEDIO  
FROM DEPARTAMENTO INNER JOIN EMPLEADO ON NUMEROD=ND  
GROUP BY NOMBRED ;
```

- h.** Salario medio de las empleadas de la EMPRESA.

```
SELECT AVG(SALARIO) AS Salario_medio_empleadas  
FROM EMPLEADO  
WHERE SEXO='M' ;
```

Solución Elmasri &Navathe 8.13 (7.18) (4)

i1. Nombre y dirección de los empleados que trabajan en algún proyecto situado en Houston pero cuyo departamento **controlador** no está situado allí.

Para que nos salga alguna solución con los datos que tenemos en la BD de prácticas podemos borrar una fila:

```
DELETE FROM LOCALIZACIONES_DEPT WHERE NUMEROD=5 AND  
LOCALIZACIOND='Houston';
```

```
SELECT NOMBRE,APELLIDO,DIRECCION  
FROM (EMPLEADO INNER JOIN TRABAJA_EN ON NSS=NSSE) INNER JOIN  
PROYECTO ON NP=NUMEROP  
WHERE LOCALIZACIONP='Houston'  
AND NUMD NOT IN (SELECT NUMEROD FROM LOCALIZACIONES_DEPT  
WHERE LOCALIZACIOND='Houston');
```

Una vez probado volver a insertar la fila borrada:

```
INSERT INTO LOCALIZACIONES_DEPT VALUES(5,'Houston');
```

Solución Elmasri & Navathe 8.13 (7.18)

- i2. Nombre y dirección de los empleados que trabajan en algún proyecto situado en Houston, pero que el departamento **para el que trabajan** no está situado allí.

```
SELECT NOMBRE,APELLIDO,DIRECCION
FROM (EMPLEADO INNER JOIN TRABAJA_EN ON NSS=NSSE) INNER JOIN
      PROYECTO ON NP=NUMEROP
WHERE LOCALIZACIONP='Houston'
AND ND NOT IN (SELECT NUMEROD FROM LOCALIZACIONES_DEPT
               WHERE LOCALIZACIOND='Houston');
```

- j. Jefes de departamento sin dependientes.

```
SELECT NOMBRE,APELLIDO
FROM EMPLEADO
WHERE NSS IN (SELECT NSS_JEFE FROM DEPARTAMENTO)
AND NSS NOT IN (SELECT NSSE FROM DEPENDIENTE);
```


Solución Elmasri &Navathe 8.16 - BD UNIVERSIDAD

- a. Nombres de los estudiantes de 1º de la carrera CS.

```
SELECT Nombre FROM ALUMNO  
WHERE Anio=1 AND Especialidad='CS';
```

- b. Nombre de los cursos impartidos por el profesor Anderson en los años 1998 y 99.

```
SELECT DISTINCT Nombre_Curso  
FROM CURSO NATURAL JOIN SECCION  
WHERE Profesor='Anderson' AND Anio IN (1998,1999);
```

Solución Elmasri &Navathe 8.16 - BD UNIVERSIDAD (2)

- c. Para cada sección impartida por el profesor Anderson, obtener el número de curso, semestre, año y número de estudiantes que tomaron la sección.

```
SELECT Identificador_Seccion ,Codigo_Curso, Semestre, Anio, COUNT(*) AS  
    Numero_alumnos  
FROM SECCION NATURAL JOIN INFORME_CALIFICACIONES  
WHERE Profesor='Anderson'  
GROUP BY Identificador_Seccion, Codigo_Curso, Semestre, Anio;
```

- d. Nombre y boletín de notas de los estudiantes de 1º de CS. El boletín incluye nombre y nº de curso, horas-crédito, semestre, año y notas de los cursos aprobados.

```
SELECT Nombre, Nombre_Curso, Codigo_Curso,  
    Creditos, Semestre, S.Anio, Calificacion
```

```
FROM ((ALUMNO AS A NATURAL JOIN INFORME_CALIFICACIONES AS I) INNER JOIN SECCION AS S  
    ON I.Identificador_seccion=S.Identificador_seccion) NATURAL JOIN CURSO AS C
```

```
WHERE A.Anio=1 AND Especialidad='CS' AND Calificacion IN ('A', 'B', 'C');
```

(se ha supuesto que la nota de APROBADO es A ,B ó C)

Porque hay un campo llamado Anio en dos de las tablas (ALUMNO Y SECCION) y no queremos que la reunión sea también por ese campo, sino solamente por el campo Identificador_Seccion

Solución Elmasri &Navathe 8.16 - BD UNIVERSIDAD (3)

- e. Nombres y departamentos de especialidad de los estudiantes calificados con la nota 'A' en todos sus cursos.

```
SELECT Nombre, Especialidad  
FROM ALUMNO
```

```
WHERE Codigo_alumno NOT IN ( SELECT Codigo_alumno  
                             FROM INFORME_CALIFICACIONES  
                             WHERE Calificacion < > 'A' );
```

Códigos de los alumnos que tienen alguna nota distinta de 'A'

- f. Nombres y departamentos de especialidad de los estudiantes que NO tengan la nota 'A' en ningún curso.

```
SELECT Nombre, Especialidad  
FROM ALUMNO
```

```
WHERE Codigo_alumno NOT IN ( SELECT Codigo_alumno  
                             FROM INFORME_CALIFICACIONES  
                             WHERE Calificacion = 'A' );
```

Códigos de los alumnos que tienen alguna 'A'