

## Ejercicio 1

Sea la base de datos de proveedores, partes y proyectos, cuyo esquema conceptual es:

- *s* (*sCod*, *sNombre*, *situacion*, *ciudad*).
- *p* (*pCod*, *pNombre*, *color*, *peso*, *ciudad*).
- *j* (*jCod*, *jNombre*, *ciudad*).
- *spj* (*sCod*, *pCod*, *jCod*, *cant*).

En los ejercicios siguientes escribir una proposición o conjunto de proposiciones en SQL para la operación indicada.

### Consultas sencillas

1. Obtener los detalles completos de todos los proyectos.

#### Solucion

```
SELECT * FROM j
```

2. Obtener los detalles completos de todos los proyectos de Londres.

#### Solucion

```
SELECT * FROM j WHERE ciudad = "Londres"
```

3. Obtener los números de los proveedores que suministran partes al proyecto J1, ordenados por número de proveedor.

#### Solucion

```
SELECT DISTINCT sCod FROM spj WHERE jCod = "J1" ORDER BY sCod
```

4. Obtener todos los envíos en los cuales la cantidad está en el intervalo de 300 a 750 inclusive.

#### Solucion

```
SELECT DISTINCT sCod FROM spj WHERE cant <= 750 AND cant >= 300
```

5. Obtener una lista de todas las combinaciones parte-color/parte-ciudad, eliminando todas las parejas color/ciudad repetidas.

#### Solucion

```
SELECT DISTINCT p.color, p2.ciudad FROM p JOIN p AS p2
```

### Reuniones

6. Obtener todas las 3-uplas número de proveedor/número de parte/ número de proyecto tales que el proveedor, la parte y el proyecto indicados estén todos en la misma ciudad (cosituados).

#### Solucion

```
SELECT DISTINCT sCod, pCod, jCod FROM s NATURAL JOIN p NATURAL JOIN j
```

7. Obtener todas las 3-uplas número de proveedor/número de parte/ número de proyecto tales que el proveedor, la parte y el proyecto indicados no estén todos cosituados.

#### Solucion

```
SELECT DISTINCT sCod, pCod, jCod  
FROM s JOIN p JOIN j  
WHERE s.ciudad <> p.ciudad OR p.ciudad <> j.ciudad
```

8. Obtener todas las 3-uplas número de proveedor/número de parte/ número de proyecto tales que el proveedor, la parte y el proyecto indicados estén todos en diferente ciudad.

### Solucion

```
SELECT DISTINCT sCod, pCod, jCod
FROM s JOIN p JOIN j
WHERE s.ciudad <> p.ciudad AND p.ciudad <> j.ciudad AND j.ciudad <> s.ciudad
```

9. Obtener los números de las partes suministradas por algún proveedor de Londres.

### Solucion

```
SELECT DISTINCT pCod FROM spj NATURAL JOIN s WHERE ciudad = "Londres"
```

10. Obtener los números de las partes suministradas por un proveedor de Londres a un proyecto en Londres.

### Solucion

```
SELECT DISTINCT pCod
FROM spj NATURAL JOIN s NATURAL JOIN j
WHERE ciudad = "Londres"
```

11. Obtener todas las parejas de nombres de ciudad tales que un proveedor de la primera ciudad suministre partes a un proyecto en la segunda ciudad.

### Solucion

```
SELECT DISTINCT s.ciudad, j.ciudad
FROM spj JOIN s JOIN j
WHERE spj.sCod = s.sCod AND spj.jCod = j.jCod
```

12. Obtener los números de las partes suministradas a un proyecto por un proveedor situado en la misma ciudad que el proyecto.

**Solucion**

```
SELECT DISTINCT pCod FROM spj NATURAL JOIN s NATURAL JOIN j
```

13. Obtener los números de los proyectos a los cuales suministra partes por lo menos un proveedor situado en una ciudad distinta.

**Solucion** COMPLETAR.

14. Obtener todas las parejas de números de parte tales que algún proveedor suministre las dos partes indicadas.

**Solucion** COMPLETAR.

**Funciones de agregados**

15. Obtener el número total de proyectos a los cuales suministra partes el proveedor S1.

**Solucion**

```
SELECT COUNT (DISTINCT jCod) FROM spj WHERE sCod = "S1"
```

16. Obtener la cantidad total de la parte P1 suministrada por el proveedor S1.

**Solucion**

```
SELECT SUM (Cant) FROM spj WHERE sCod = "S1" AND pCod = "P1"
```

17. Para cada parte suministrada a un proyecto, obtener el número de parte, el número de proyecto y la cantidad total correspondiente.

**Solucion**

```
SELECT pCod, jCod, SUM (Cant) FROM spj GROUP BY pCod, jCod
```

18. Obtener los números de las partes suministradas a algún proyecto tales que la cantidad promedio suministrada sea mayor que 320.

**Solucion**

```
SELECT pCod FROM spj GROUP BY pCod HAVING AVG(cant) > 320
```

**Diversas**

19. Obtener todos los envíos para los cuales la cantidad no sea nula.

**Solucion**

```
SELECT * FROM spj WHERE cant <> 0
```

20. Obtener números de proyecto y ciudades en los cuales la segunda letra del nombre de la ciudad sea una "o".

**Solucion**

```
SELECT DISTINCT jCod, ciudad FROM j WHERE ciudad LIKE "_o%"
```

### Subconsultas

21. Obtener los nombres de los proyectos a los cuales suministra partes el proveedor S1.

#### Solucion

```
SELECT DISTINCT jNombre FROM spj NATURAL JOIN j WHERE sCod = "S1"
```

22. Obtener los colores de las partes suministradas por el proveedor S1.

#### Solucion

```
SELECT DISTINCT color FROM spj NATURAL JOIN p WHERE sCod = "S1"
```

23. Obtener los números de las partes suministradas a cualquier proyecto en Londres.

#### Solucion

```
SELECT DISTINCT pCod FROM spj NATURAL JOIN j WHERE ciudad = "Londres"
```

24. Obtener los números de los proyectos donde se utilice al menos una de las partes suministradas por el proveedor S1.

#### Solucion

```
SELECT DISTINCT jCod FROM spj NATURAL JOIN j  
WHERE pCod IN (SELECT pCod FROM spj WHERE sCod = "S1")
```

25. Obtener los números de los proveedores que suministren por lo menos una de las partes suministradas por al menos uno de los proveedores que suministran por lo menos una parte roja.

#### Solucion

 COMPLETAR.

26. Obtener los números de los proveedores cuya situación sea inferior a la del proveedor S1.

**Solucion**

```
SELECT sCod FROM s
WHERE situacion < (SELECT situacion FROM s WHERE sCod = "S1")
```

27. Obtener los números de los proyectos cuya ciudad sea la primera en la lista alfabética de las ciudades donde hay proyectos.

**Solucion**

```
SELECT jCod FROM j
WHERE ciudad = (SELECT ciudad FROM j ORDER BY ciudad LIMIT 1)
```

**EXISTS**

28. Repetir el ejercicio 23 utilizando EXISTS en la solución.

**Solucion**

```
SELECT DISTINCT pCod FROM p WHERE EXISTS
  (SELECT * FROM spj NATURAL JOIN j
   WHERE spj.pCod = p.pCod AND ciudad = "Londres")
```

29. Repetir el ejercicio 24 utilizando EXISTS en la solución.

**Solucion**

```
SELECT DISTINCT jCod FROM j WHERE EXISTS
  (SELECT * FROM spj WHERE spj.jCod = j.jCod AND sCod = "S1")
```

30. Obtener los números de los proyectos a los cuales no suministra ninguna parte roja ninguno de los proveedores de Londres.

**Solucion**

```
SELECT DISTINCT jCod FROM j WHERE NOT EXISTS
  (SELECT * FROM spj JOIN p JOIN s
   WHERE spj.jCod = j.jCod AND
         spj.pCod = p.pCod AND
         spj.sCod = s.sCod AND
         color = "Rojo" AND
         s.ciudad = "Londres")
```

31. Obtener los números de los proyectos para los cuales S1 es el único proveedor.

**Solucion** COMPLETAR.

**Union**

32. Construir una lista ordenada de todas las ciudades en las cuales esté situado por lo menos un proveedor, una parte o un proyecto.

**Solucion**

```
SELECT DISTINCT ciudad FROM
  (SELECT ciudad FROM j UNION
   SELECT ciudad FROM s UNION
   SELECT ciudad FROM p) AS temp
ORDER BY ciudad
```

**Operaciones de actualizacion**

33. Cambiar a gris el color de todas las partes rojas.
34. Eliminar todos los proyectos para los cuales no haya envíos.
35. Insertar un nuevo proveedor (S10) en la tabla S. El nombre y la ciudad son Salazar y Nueva York, respectivamente; la situación no se conoce todavía.