

1. Dadas las tablas ALUM y NUEVOS, insertar en la tabla ALUM los nuevos alumnos.

```
SQL> INSERT INTO alum
      (SELECT * FROM nuevos
      MINUS
      SELECT * FROM alum);
```

2. Borrar de la tabla ALUM los ANTIGUOS alumnos.

```
DELETE FROM alum
WHERE nombre IN(SELECT nombre FROM antiguos
                INTERSECT
                SELECT nombre FROM alum)
```

2 filas borradas.

3. EMPLA y DEPART. Insertar un empleado de apellido 'saavedra' con número 200, la fecha de alta será la actual, el SALARIO será el mismo salario de 'sala' más el 20 por 100 y el resto de datos serán los mismos que los datos de 'sala'.

```
INSERT INTO emple
SELECT 200, 'SAAVEDRA', oficio, dir, sysdate, salario+salario*20/100, comision, dept_no
FROM emple
WHERE apellido='SALA'
```

1 fila creada.

4. Modificar el número de departamento de 'saavedra'. El nuevo departamento será el departamento donde hay más empleados cuyo oficio sea 'empleado'.

```
UPDATE emple
SET dept_no=(SELECT dept_no FROM emple
             WHERE oficio='EMPLEADO'
             GROUP BY dept_no
             HAVING COUNT(*)=(SELECT MAX(COUNT(*)) FROM emple
                               WHERE oficio='EMPLEADO'
                               GROUP BY dept_no))
WHERE apellido='SAAVEDRA'
```

1 fila actualizada.

5. Borrar todos los departamentos de la tabla DEPART para los cuales no existan empleados en emple

```
SQL> DELETE FROM depart
      WHERE dept_no=(SELECT dept_no FROM emple
                     GROUP BY dept_no
                     HAVING COUNT(*)=0);
```

0 filas borradas.

6. PERSONAL, PROFESORES y CENTROS. Modificar el número de plazas con un valor igual a la mitad en aquellos centros con menos de dos profesores.

```
SQL> UPDATE centros
      SET num_plazas=num_plazas/2
      WHERE cod_centro IN (SELECT cod_centro FROM profesores
                           GROUP BY cod_centro
                           HAVING COUNT(*)<2);
```

2 filas actualizadas.

7. Eliminar los centros que no tengan personal.

```
SQL> DELETE FROM centros
      WHERE cod_centro IN (SELECT cod_centro FROM profesores
                           GROUP BY cod_centro
                           HAVING COUNT(*) =0)
```

0 filas borradas.

8. Añadir un nuevo profesor en el centro o en los centros cuyo número de administrativos sea 1 en la especialidad de 'idioma', con dni 8790055 y de nombre 'Clara Salas'.

```
INSERT INTO profesores
(cod_centro, dni, apellidos, especialidad)
SELECT cod_centro, 8790055, 'CLARA SALAS', 'IDIOMA'
FROM centros
WHERE cod_centro IN (SELECT cod_centro FROM personal
                     WHERE funcion='ADMINISTRATIVO'
                     GROUP BY cod_centro
                     HAVING COUNT(*)=1)
```

2 filas creadas.

9. Borrar el personal que esté en centros de menos de 300 plazas y con menos de dos profesores.

```
SQL> DELETE FROM personal
      WHERE cod_centro IN (SELECT cod_centro FROM centros
                           WHERE num_plazas < 300 AND cod_centro IN
                               (SELECT cod_centro FROM profesores
                                GROUP BY cod_centro
                                HAVING COUNT(*)<2));
```

2 filas borradas.

10. Borrar a los profesores que estén en la tabla PROFESORES y que no estén en la tabla PERSONAL.

```
SQL> DELETE FROM profesores
      WHERE dni NOT IN (SELECT dni FROM personal
                       WHERE funcion='PROFESOR');
```

3 filas borradas.

11. Las tablas ARTÍCULOS,FABRICANTES,TIENDAS,PEDIDOS Y VENTAS vienen determinadas por los atributos NOT NULL.

Dar de alta un nuevo artículo de 'Primera' categoría para los fabricantes de 'francia' y abastecer con cinco unidades de ese artículo a todas las tiendas y en la fecha de hoy.

```
SQL> INSERT INTO articulos
```

```
SELECT 'Chocolate',cod_fabricante,1,'Primera',100,75,35
FROM fabricantes
WHERE pais='FRANCIA';
```

1 fila creada.

12. Insertar un pedido de 20 unidades en la tienda '1111-a' con el artículo que mayor número de ventas haya realizado.

```
SQL> INSERT INTO ventas
SELECT '1111-A', articulo, cod_fabricante, peso, categoria, SYSDATE, 20
FROM ventas
WHERE articulo=(SELECT articulo FROM ventas
GROUP BY articulo
HAVING SUM(unidades_vendidas) =
(SELECT (MAX(SUM(unidades_vendidas)))
FROM ventas
GROUP BY articulo));
```

4 filas creadas.

13. Dar de alta una tienda en la provincia de Madrid y abastecerla con 20 unidades de cada uno de los artículos existentes.

```
INSERT INTO tiendas
VALUES ('6755-J','Mercancias Andres','C/nose 32','Zaragoza','Zaragoza',50012)
```

1 fila creada.

```
SQL> INSERT INTO pedidos
SELECT '6755-J', articulo, cod_fabricante, peso, categoria, SYSDATE, 20
FROM articulos
```

31 filas creadas.

14. Dar de alta dos tiendas en la provincia de Sevilla y abastecerlas con 30 unidades de artículos de la marca de fabricante gallo.

```
SQL> INSERT INTO tiendas
VALUES('2349-D','OSCAR MARKET','C/nOSE 34','SEVILLA','SEVILLA',41253)
```

1 fila creada.

```
SQL> INSERT INTO tiendas
VALUES('1239-A','HECTOIS MARKET','C/nOSE 34','SEVILLA','SEVILLA',41253)
```

1 fila creada.

```
SQL> INSERT INTO pedidos
SELECT nif, articulo, cod_fabricante, peso, categoria, SYSDATE, 30
FROM articulos, tiendas
WHERE tiendas.provincia='SEVILLA'
AND cod_fabricante = (SELECT cod_fabricante FROM fabricantes
WHERE nombre='GALLO')
```

15. Realizar una venta para todas las tiendas de toledo de 10 unidades en los artículos de primera categoría.

```
SQL> INSERT INTO ventas
      SELECT nif,articulo,cod_fabricante,peso,categoria,SYSDATE,10
      FROM articulos,tiendas
      WHERE categoria='Primera'
            AND provincia='TOLEDO';
```

36 filas creadas.

16. Para aquellos artículos de los que se hayan vendido más de 30 unidades, realizar un pedido de 10 unidades para la tienda con NIF '5555-B' con la fecha actual.

```
SQL> INSERT INTO pedidos
      SELECT '555-D',articulo,cod_fabricante,peso,categoria,SYSDATE,10
      FROM ventas
      WHERE articulo IN (SELECT articulo FROM ventas
                        GROUP BY articulo
                        HAVING SUM(unidades_vendidas)>30)
```

46 filas creadas.

17. Cambiar todos los artículos de 'primera' categoría del país ITALIA

```
SQL> UPDATE articulos
      SET categoria='Quinta'
      WHERE categoria='Primera' AND
            cod_fabricante = (SELECT cod_fabricante FROM fabricantes
                              WHERE pais='ITALIA');
```

2 filas actualizadas.

18. Eliminar aquellas tiendas que no han realizado ventas.

```
1 DELETE FROM TIENDAS
2* WHERE NIF NOT IN (SELECT NIF FROM VENTAS)
```

3 filas borradas.

19. Eliminar los artículo que no hayan tenido no compras ni ventas.

```
SQL> R
1 DELETE FROM ARTICULOS
2 WHERE ARTICULO NOT IN (SELECT ARTICULO FROM VENTAS
3                        GROUP BY ARTICULO)
4 AND ARTICULO NOT IN (SELECT ARTICULO FROM PEDIDOS
5*                     GROUP BY ARTICULO)
```

4 filas borradas.

20. Borrar los pedidos de 'primera' categoría cuyo país de procedencia sea BÉLGICA.

```
SQL> DELETE FROM PEDIDOS
2 WHERE COD_FABRICANTE = (SELECT COD_FABRICANTE FROM FABRICANTES)
```

3 WHERE PAIS='BELGICA');

6 filas borradas.

21. Borrar los pedidos que no tengan tienda.

**SQL> DELETE FROM PEDIDOS
2 WHERE NIF NOT IN(SELECT NIF FROM TIENDAS);**

3 filas borradas.

22. Restar uno a las unidades de los últimos pedidos de la tienda con nif 555-B

**1 UPDATE PEDIDOS
2 SET UNIDADES_PEDIDAS=UNIDADES_PEDIDAS-1
3 WHERE NIF='5555-B' AND
4 FECHA_PEDIDO=
5* (SELECT MAX(FECHA_PEDIDO) FROM PEDIDOS WHERE NIF='5555-B')**

1 fila actualizada.