TAREFAS PROPOSTAS . SOLUCIÓNS

1. Tarefa de consultas simples

- Consultas propostas na **BD SOCIEDADE_CULTURAL**.

```
--Seleccionamos a BD SOCIEDADE_CULTURAL
USE SOCIEDADE_CULTURAL;
/*** CONSULTA PROPOSTA 1 ***/
--Nome e apelidos (cada un nunha columna)
--de todos os empregados
SELECT nome, ape1, ape2
FROM EMPREGADO;

/*** CONSULTA PROPOSTA 2 ***/
--Número, nome e apelidos (cada un nunha columna)
--de todos os empregados
SELECT numero, nome, ape1, ape2
FROM EMPREGADO;

/*** CONSULTA PROPOSTA 3 ***/
--Número, nome e apelidos (cada un nunha columna)
--de todos os empregados por orde alfabética de
--apelidos e nome
SELECT numero, nome, ape1, ape2
FROM EMPREGADO
ORDER BY ape1, ape2, nome;

/*** CONSULTA PROPOSTA 4 ***/
--Número, nome e apelidos (cada un nunha columna)
--de todos os empregados por orde alfabética de
--apelidos e nome. Os nomes das columnas no resultado serán:
--num_socio, nome_socio, apelido1, apelido2
SELECT numero as num_socio, nome as nome_socio,
       ape1 as apelido1, ape2 as apelido2
FROM EMPREGADO
ORDER BY ape1, ape2, nome;

/*** CONSULTA PROPOSTA 5 ***/
--Número, nome completo (os 4 campos nunha única columna,
--de nome socio, co formato numero - ape1 ape2, nome)
--e salario de todos os empregados.
--No resultado deberán aparecer primeiro os que máis cobran.
SELECT cast(numero as varchar(7))+' - '+ape1+' '+ape2+', '+ nome
       as socio,
         salario_mes
FROM EMPREGADO
ORDER BY salario_mes desc;
```

```
/*** CONSULTA PROPOSTA 6 ***/
--Número, nome completo (os 4 campos nunha única columna,
--de nome socio, co formato numero - ape1 ape2, nome)
--e salario de todo o profesorado.
--No resultado deberán aparecer primeiro os
--que máis cobran. O campo cargo contén PRF para
--o profesorado, e ADM se é un ou unha administrativo.
SELECT cast(numero as varchar(7))+' - '+ape1+' '+ape2+', '+ nome
       as socio,
         salario_mes
FROM EMPREGADO
WHERE cargo='PRF'
ORDER BY salario_mes desc;

/*** CONSULTA PROPOSTA 7 ***/
--Número identificador do profesorado que imparte clases.
--Como é lóxico, se un profesor imparte máis dunha
--actividade, o seu número só pode aparecer unha vez.
SELECT DISTINCT num_profesorado_imparte
FROM ACTIVIDADE;

/*** CONSULTA PROPOSTA 8 ***/
--Número identificador das actividades ás
--que asiste profesorado, é dicir,
--cursadas por profesorado.
SELECT DISTINCT id_actividade
FROM PROFE_CURSA_ACTI;

/*** CONSULTA PROPOSTA 9 ***/
--Nome, prezo, e prezo rebaixado un 20%,
--da actividade de nome XADREZ.
--Solución1
SELECT nome, prezo, prezo-(prezo*0.20) as prezo_rebaixado
FROM ACTIVIDADE
WHERE nome='XADREZ';
--Solución2
SELECT nome, prezo, prezo*0.80 as prezo_rebaixado
FROM ACTIVIDADE
WHERE nome='XADREZ';

/*** CONSULTA PROPOSTA 10 ***/
--NIF, nome e apelidos dos socios
--dos que non temos teléfono gardado.
SELECT NIF, nome, ape1, ape2
FROM SOCIO
WHERE telefono1 IS NULL AND
      telefono2 IS NULL;
```

```
/*** CONSULTA PROPOSTA 11 ***/
--NIF, nome e apelidos e data de nacemento dos socios
--nados entre 1980 e 1990 (ambos incluídos).
SELECT NIF, nome, ape1, ape2, data_nac
FROM SOCIO
WHERE data_nac BETWEEN '1/1/1980' AND '31/12/1990';

/*** CONSULTA PROPOSTA 12 ***/
--Todos os datos das actividades cuxo nome contén a letra T.
SELECT *
FROM ACTIVIDADE
WHERE nome LIKE '%T%';

/*** CONSULTA PROPOSTA 13 ***/
--Nome e importe das cotas cun custo de 30 ou 100 euros.
--Solución1
SELECT nome, importe
FROM COTA
WHERE importe IN (30,100);
--Solución2
SELECT nome, importe
FROM COTA
WHERE importe=30 OR
        importe=100;

/*** CONSULTA PROPOSTA 14 ***/
--Nome e número de prazas das actividades
--que non teñen nin 15 nin 20 prazas.
--Solución1
SELECT nome, num_prazas
FROM ACTIVIDADE
WHERE num_prazas NOT IN (15,20);
--Solución2
SELECT nome, num_prazas
FROM ACTIVIDADE
WHERE num_prazas!=15 AND
      num_prazas!=20;
```

- Consultas propostas na **BD EMPRESA**.

```
/*** CONSULTA PROPOSTA 15 ***/
--Nome de todos os clientes por orde alfabética.
--Fíxate como se chama o campo que garda o nome do cliente.
SELECT nome
FROM CLIENTE
ORDER BY nome;
```

```
/*** CONSULTA PROPOSTA 16 ***/
--Nome das rexións nas que ten sucursais a empresa.
SELECT DISTINCT rexion
FROM SUCURSAL;

/*** CONSULTA PROPOSTA 17 ***/
--Identificador dos produtos que nos pediron
--nalgún momento. No resultado debe aparecer
--nunha soa columna o código do fabricante e o
--identificador do produto separados por un guión.
SELECT DISTINCT cod_fabricante+'-'+id_produto as produtos
FROM PEDIDO;

/*** CONSULTA PROPOSTA 18 ***/
--Información completa das sucursais non
--dirixidas polo empregado número 108.
--Solución1
SELECT identificador, cidade, rexion,
       num_empregado_director, obxectivo
FROM SUCURSAL
WHERE num_empregado_director!=108;
--Solución2
SELECT *
FROM SUCURSAL
WHERE num_empregado_director!=108;

/*** CONSULTA PROPOSTA 19 ***/
--Nome e límite de crédito do cliente número 1107.
SELECT nome, limite_de_credito
FROM CLIENTE
WHERE numero=1107;

/*** CONSULTA PROPOSTA 20 ***/
--Número e data dos pedidos feitos entre o 1 de agosto
--e o 31 de decembro de 2014. Só debe aparecer a data
--de cada pedido, sen a hora, co formato dd-mm-aaaa.
--Deben aparecer primeiro no resultado os pedidos máis recentes.
--Para resolver esta consulta non se poden utilizar
--operadores de comparación (>, <, >=, <=, < >, !=).
SELECT numero, convert(char(10), data_pedido, 105)
       as data_do_pedido
FROM PEDIDO
WHERE data_pedido BETWEEN '1-08-2014' AND '31-12-2014'
ORDER BY data_pedido DESC;
```

```
/*** CONSULTA PROPOSTA 21 ***/
--Código e nome dos fabricantes cuxo nome
--ten por segunda letra O.
SELECT codigo, nome
FROM FABRICANTE
WHERE nome LIKE '_O%';

/*** CONSULTA PROPOSTA 22 ***/
--Descrición e prezo dos produtos dos
--que non temos existencias.
SELECT descricion, prezo
FROM PRODUTO
WHERE existencias=0;

/*** CONSULTA PROPOSTA 23 ***/
--Número identificador e nome completo dos
--empregados que non teñen xefe.
SELECT numero, nome, ape1, ape2
FROM EMPREGADO
WHERE num_empregado_xefe IS NULL;

/*** CONSULTA PROPOSTA 24 ***/
--Descrición e unidades existentes, dos produtos
--con existencias maiores
--de 10 unidades e menores de 100. Para resolver esta
--consulta non se poden utilizar operadores
--de comparación (>, <, >=, <=, < >, !=).
SELECT descricion, existencias
FROM PRODUTO
WHERE existencias BETWEEN 11 AND 99;
```

2. Tarefa de consultas resumo

- Consultas propostas na **BD EMPRESA**.

```
--Seleccionamos a BD EMPRESA
USE EMPRESA;

/*** CONSULTA PROPOSTA 1 ***/
--Media de unidades vendidas por vendedor.
SELECT num_empregado, avg(cantidade) as media_unidades
FROM PEDIDO
GROUP BY num_empregado;
```

```
/*** CONSULTA PROPOSTA 2 ***/
--Prezo máis barato de produto, prezo máis caro,
--prezo medio, suma de todos os prezos
--e número de produtos distintos existentes.
SELECT  min(prezo) as prezo_inferior,
        max(prezo) as prezo_superior,
             avg(prezo) as prezo_medio,
             sum(prezo) as prezo_total,
             count(identificador) as total_produtos
FROM PRODUTO;

/*** CONSULTA PROPOSTA 3 ***/
--Número de pedidos realizados polo cliente 1103
SELECT count(numero) as num_pedidos_cliente1103
FROM PEDIDO
WHERE num_cliente=1103;

/*** CONSULTA PROPOSTA 4 ***/
--Número de pedidos realizados por cada cliente.
SELECT num_cliente, count(numero) as num_pedidos
FROM PEDIDO
GROUP BY num_cliente;

/*** CONSULTA PROPOSTA 5 ***/
--Número de pedidos realizados por cada cliente
--só para os clientes que fixeran máis de 2 pedidos.
SELECT count(numero) as
FROM PEDIDO
GROUP BY cod_fabricante
HAVING count(numero)>2;

/*** CONSULTA PROPOSTA 6 ***/
--Número de pedidos realizados por cada cliente
--só para os clientes que fixeran máis de 2 pedidos
--e que ademais teñen unha media de unidades
--mercadas (cantidade) inferior a 10.
SELECT num_cliente, count(numero) as num_pedidos
FROM PEDIDO
GROUP BY num_cliente
HAVING count(numero)>2 AND
       avg(cantidade)<10;

/*** CONSULTA PROPOSTA 7 ***/
--Cantidade total de sucursais que hai por rexión.
--Aparecerá o nome da rexión e na mesma columna
--separado por un guión, a cantidade de sucursais
--situadas nesa rexión.
SELECT rexion+' - '+ cast(count(*) as varchar(5)) as total_sucursais
FROM SUCURSAL
GROUP BY rexion;
```

3. Tarefa de consultas con combinacións

- Consultas propostas na **BD EMPRESA**.

```
--Seleccionamos a BD EMPRESA
USE EMPRESA;

/*** CONSULTA PROPOSTA 1 ***/
--Nome de todos os fabricantes dos que se fixeron pedidos.
--Solución1
SELECT DISTINCT f.nome
FROM FABRICANTE f, PEDIDO p
WHERE f.codigo=p.cod_fabricante;
--Solución2
SELECT DISTINCT f.nome
FROM FABRICANTE f INNER JOIN PEDIDO p
     ON f.codigo=p.cod_fabricante;

/*** CONSULTA PROPOSTA 2 ***/
--Nome de todos os fabricantes con ou sen pedidos.
--Se tiveron pedidos aparecerá o número de pedido.
--Se non ten pedidos aparecerá como número de pedido 99.
SELECT f.nome, ISNULL(p.numero,99) as num_pedido
FROM FABRICANTE f LEFT JOIN PEDIDO p
     ON f.codigo=p.cod_fabricante;

/*** CONSULTA PROPOSTA 3 ***/
--Nome de todos os fabricantes con ou sen pedidos.
--Se tiveron pedidos aparecerá o número de pedido.
--Se non ten pedidos aparecerá 'Sen pedidos'.
SELECT f.nome, ISNULL(cast(p.numero as varchar(11)),'Sen pedidos')
       as num_pedido
FROM FABRICANTE f LEFT JOIN PEDIDO p
     ON f.codigo=p.cod_fabricante;

/*** CONSULTA PROPOSTA 4 ***/
--Código dos produtos (cod_fabricante-id_produto) e
--descrición dos produtos que non
--foron pedidos nunca.
SELECT pr.cod_fabricante+'-'+pr.identificador as id_produto,
          pr.descricion
FROM PRODUTO pr LEFT JOIN PEDIDO pe
     ON pr.cod_fabricante= pe.cod_fabricante AND
          pr.identificador =pe. id_produto
WHERE pe.numero is NULL;
```

```
/*** CONSULTA PROPOSTA 5 ***/
--Produto cartesiano entre a táboa de
--sucursais e a de empregados.
--Solución1
SELECT s.cidade,
       e.nome+' '+e.ape1+' '+isnull(e.ape2,'') as empregado
FROM SUCURSAL s, EMPREGADO e;
--Solución2
SELECT s.cidade,
       e.nome+' '+e.ape1+' '+isnull(e.ape2,'') as empregado
FROM SUCURSAL s CROSS JOIN EMPREGADO e;

/*** CONSULTA PROPOSTA 6 ***/
--Número e nome completo de todos os empregados,
--así como a cidade da sucursal que dirixen.
--Nos empregados que non son directores de sucursais,
--aparecerá 'Non é director.'.
SELECT e.numero,
       e.nome+' '+e.ape1+' '+isnull(e.ape2,'') as nome,
         isnull(s.cidade, 'Non é director.') as sucursal
FROM EMPREGADO e LEFT JOIN SUCURSAL s
     ON e.numero=s.num_empregado_director;

/*** CONSULTA PROPOSTA 7 ***/
--Número e nome completo dos empregados que teñen xefe,
--co número do seu xefe nunha segunda columna.
SELECT cast(em.numero as varchar(5))+'-'+
       em.nome+' '+em.ape1+' '+isnull(em.ape2,'') as empregado,
         cast(xefe.numero as varchar(5))+'-'+
       xefe.nome+' '+xefe.ape1+' '+isnull(xefe.ape2,'') as xefe
FROM EMPREGADO em, EMPREGADO xefe
WHERE em.num_empregado_xefe=xefe.numero;

/*** CONSULTA PROPOSTA 8 ***/
--Número e nome completo dos empregados que teñen xefe,
--co número do seu xefe nunha segunda columna.
SELECT cast(em.numero as varchar(5))+'-'+
       em.nome+' '+em.ape1+' '+isnull(em.ape2,'') as empregado,
         isnull(cast(xefe.numero as varchar(5))+'-'+
       xefe.nome+' '+xefe.ape1+' '+isnull(xefe.ape2,''),'Xefe por
designar.') as xefe
FROM EMPREGADO em LEFT JOIN EMPREGADO xefe
     ON em.num_empregado_xefe=xefe.numero;
```

```
/*** CONSULTA PROPOSTA 9 ***/
--Nome completo de todos os empregados co
--nome do cliente que teñen asignado.
SELECT  isnull(e.nome+' '+e.ape1+' '+isnull(e.ape2,''),'Sen vendedor.')
        as empregado,
        isnull(c.nome, 'Sen cliente.') as nome_cliente
FROM EMPREGADO e FULL JOIN CLIENTE c
     ON e.numero=c.num_empregado_asignado;

/*** CONSULTA PROPOSTA 10 ***/
--Escolle unha das túas solucións das consultas propostas
--nas que empregaches un LEFT JOIN, e modifícaa usando RIGHT JOIN.
--Solución consulta proposta 2 con RIGHT JOIN
SELECT f.nome, ISNULL(p.numero,99) as num_pedido
FROM  PEDIDO p RIGHT JOIN FABRICANTE f
     ON f.codigo=p.cod_fabricante;
```

4. Tarefa de consultas con subconsultas

- Consultas propostas na BD EMPRESA.

```
--Seleccionamos a BD EMPRESA
USE EMPRESA;
/*** CONSULTA PROPOSTA 1 ***/
--Nome de todos os fabricantes dos que hai
--produtos na BD. Non se permite usar
--combinacións nesta consulta.
SELECT nome
FROM FABRICANTE
WHERE codigo IN (SELECT cod_fabricante
                 FROM PRODUTO);

/*** CONSULTA PROPOSTA 2 ***/
--Nome de todos os fabricantes dos que non hai
--produtos na BD.
--Non se permite usar combinacións nesta consulta.
SELECT nome
FROM FABRICANTE
WHERE codigo NOT IN (SELECT cod_fabricante
                     FROM PRODUTO);

/*** CONSULTA PROPOSTA 3 ***/
--Número de pedido, cantidade e data de pedido para
--aqueles pedidos recibidos nos días en que un novo
--empregado foi contratado.
SELECT numero, cantidade, data_pedido
FROM PEDIDO
WHERE data_pedido IN (SELECT data_contrato
                      FROM EMPREGADO);
```

```
/*** CONSULTA PROPOSTA 4 ***/
--Cidade e obxectivo das sucursais cuxo obxectivo
--supera a media das cotas de todos os vendedores da BD.
SELECT cidade, obxectivo
FROM SUCURSAL
WHERE obxectivo > (SELECT avg(cota_de_vendas)
                   FROM EMPREGADO);

/*** CONSULTA PROPOSTA 5 ***/
--Número de empregado e cantidade media dos pedidos
--daqueles empregados cuxa cantidade media de pedido
--é superior á cantidade media global (de todos os pedidos).
SELECT num_empregado, avg(cantidade) as cant_media
FROM PEDIDO
GROUP BY num_empregado
HAVING avg(cantidade) > (SELECT avg(cantidade)
                         FROM PEDIDO);

/*** CONSULTA PROPOSTA 6 ***/
--Nome dos clientes que aínda non fixeron pedidos.
SELECT nome
FROM CLIENTE
WHERE numero NOT IN (SELECT num_cliente
                     FROM PEDIDO);

/*** CONSULTA PROPOSTA 7 ***/
--Nome completo dos empregados cuxas cotas son iguais
--ou superiores ao obxectivo da sucursal da cidade de Vigo.
--Ten en conta que se a cota dun vendedor (empregado) é nula
--debemos considerala como un 0, e do mesmo xeito actuaremos
--co obxectivo da sucursal.
SELECT nome, ape1, ape2
FROM EMPREGADO
WHERE isnull(cota_de_vendas,0)>=(SELECT isnull(obxectivo,0)
                                                    FROM SUCURSAL
                                                    WHERE cidade='Vigo');

/*** CONSULTA PROPOSTA 8 ***/
--Nome dos produtos para os que existe polo menos un pedido
--que ten unha cantidade de polo menos 20 unidades. A solución deberá
--facerse empregando o predicado EXISTS cunha subconsulta
--correlacionada.
SELECT pr.descricion
FROM PRODUTO pr
WHERE EXISTS (SELECT p.numero
              FROM PEDIDO p
                     WHERE p.cantidade >=20 AND
                           p.cod_fabricante=pr.cod_fabricante AND
                                p.id_produto=pr.identificador);
```

```
/*** CONSULTA PROPOSTA 9 ***/
--Cidades das sucursais onde exista algún empregado cuxa
--cota de ventas represente máis do 80% do obxectivo da
--oficina onde traballa. Para resolver esta consulta deberase
--empregar unha subconsulta correlacionada precedida de ANY.
SELECT s.cidade
FROM SUCURSAL s
WHERE 0.80*s.obxectivo<ANY(SELECT e.cota_de_ventas
                                FROM EMPREGADO e
                                WHERE
e.id_sucursal_traballa=s.identificador);

/*** CONSULTA PROPOSTA 10 ***/
--Nome dos clientes cuxos empregados asignados traballan en
--sucursais da rexión OESTE. Non se poden usar joins,
--só subconsultas encadeadas.
SELECT nome
FROM CLIENTE
WHERE num_empregado_asignado IN (SELECT numero
                                 FROM EMPREGADO
                                 WHERE id_sucursal_traballa IN
                                                (SELECT identificador
                                                 FROM SUCURSAL
                                                 WHERE rexion='OESTE'));
```

5. Tarefa de consultas con funcións integradas no xestor

- Consultas propostas na **BD EMPRESA**.

```
--Seleccionamos a BD EMPRESA
USE EMPRESA;

/*** CONSULTA PROPOSTA 1 ***/
--Código ASCII da vogal E maiúscula.
SELECT 'E' as vogal, ascii('E') as codigo_asciiE;

/*** CONSULTA PROPOSTA 2 ***/
--Carácter que lle corresponde aos
--códigos ASCII: 70, 80, 90.
SELECT char(70) as caracter1,
       char(80) as caracter2,
         char(90) as caracter10;
```

```
/*** CONSULTA PROPOSTA 3 ***/
SELECT 'O empregado con nome '+nome+' '+ape1+rtrim(' '+isnull(ape2,''))+
       ' ten que acadar unha cota de vendas de '+
         cast(cota_de_vendas as varchar(12))
         as "Empregados e cotas"
FROM EMPREGADO;

/*** CONSULTA PROPOSTA 4 ***/
--Datas nas que se contrataron empregados
SELECT DISTINCT convert(char(10), data_contrato, 105)
       as "Datas de contratación"
FROM EMPREGADO;

/*** CONSULTA PROPOSTA 5 ***/
--Tres primeiros caracteres da cidade, os dous últimos da
--rexión e separado por un guión baixo, o número de
--caracteres do nome da cidade
SELECT left(cidade,3)+right(rexion,2)+'-'+
       cast(len(cidade) as varchar(15)) as Abrev_sucursais
FROM SUCURSAL;

/*** CONSULTA PROPOSTA 6 ***/
--Segundo carácter do código do fabricante en minúscula
--máis o terceiro, cuarto, quinto e sexto da descrición do produto.
--En minúsculas, e nunha segunda columna en maiúsculas.
SELECT lower(substring(cod_fabricante,2,1)+substring(descricion,3,4))
       as abrev_produto_minusc,
         upper(substring(cod_fabricante,2,1)+substring(descricion,3,4))
       as abrev_produto_maiusc
FROM PRODUTO;

/*** CONSULTA PROPOSTA 7 ***/
--Nomes dos empregados co formato ape1 ape2, nome
SELECT rtrim(ape1+' '+isnull(ape2,''))+', '+nome as Empregados
FROM EMPREGADO;

/*** CONSULTA PROPOSTA 8 ***/
--Distintos títulos dos nosos empregados en castelán.
SELECT DISTINCT replace(titulo, 'VENDAS', 'VENTAS') as titulo
FROM EMPREGADO;
```

```
/*** CONSULTA PROPOSTA 9 ***/
--Consulta que devolva información de tempo.
SELECT getdate() as data_hora_sen_axuste,
       sysdatetimeoffset() as data_hora_con_axuste,
         month(getdate()) as mes_en_numero1,
         datepart(month, getdate()) as mes_en_numero2,
         year(getdate()) as ano_actual,
         datename(month, getdate()) as nome_mes,
         datepart(hour,getdate()) as hora_actual,
         datepart(nanosecond, sysdatetime()) as nanosegundos_actual;

/*** CONSULTA PROPOSTA 10 ***/
--Empregados, data contrato e data contrato adiantada un ano.
SELECT nome, ape1, isnull(ape2,'') as ape2,
         convert(char(10),data_contrato,103) as data_contrato,
       convert(char(10),dateadd(year,-1,data_contrato),103)
         as data_adiantada1ano
FROM EMPREGADO;

/*** CONSULTA PROPOSTA 11 ***/
--Pedidos, data e data de pedido retrasada dous meses.
SELECT numero,
         convert(char(10),data_pedido,105) as data_pedido,
       convert(char(10),dateadd(month,2,data_pedido),105)
         as data_retrasada2meses
FROM PEDIDO;

/*** CONSULTA PROPOSTA 12 ***/
--Nome e apelido dos empregados, data de contrato
--e número de anos que leva traballando.
SELECT nome, ape1, isnull(ape2,'') as ape2, data_contrato,
       datediff(year, data_contrato, getdate()) as anos_traballando
FROM EMPREGADO;

/*** CONSULTA PROPOSTA 13 ***/
--Descrición dos produtos, prezo, prezo por defecto, por exceso,
--raíz cadrada, cadrado e cubo do prezo.
SELECT descricion, prezo,
       floor(prezo) as prezo_por_defecto,
         ceiling(prezo) as prezo_por_exceso,
         sqrt(prezo) as raiz_cadrada_do_prezo,
         square(prezo) as cadrado_do_prezo,
         power(prezo, 3) as cubo_do_prezo
FROM PRODUTO;
```

```
/*** CONSULTA PROPOSTA 14 ***/
--Descrición dos produtos, prezo e a
--raíz cadrada con 4 enteiros e 3 decimais.
SELECT descricion, prezo,
       convert(numeric(7,3),sqrt(prezo)) as raiz_cadrada_do_prezo
FROM PRODUTO;


/*** CONSULTA PROPOSTA 15 ***/
--Información do servidor: idioma, número máximo de
--conexións permitidas, nome do servidor e versión.
SELECT @@language as idioma,
       @@max_connections as conexions_maximas,
         @@servername as nome_servidor,
         @@version as version;

/*** CONSULTA PROPOSTA 16 ***/
--Descrición e existencias dos produtos. Na terceira columna:
--Suficientes se existencias son máis de 20,
--insuficientes se son menos ou iguais a 20.
--Solución1
SELECT descricion, existencias,
       case
             when existencias > 20 then 'Suficientes'
             else 'Insuficientes'
         end as estado_existencias
FROM PRODUTO;
--Solución2
SELECT descricion, existencias,
       iif(existencias > 20,'Suficientes','Insuficientes')
         as estado_existencias
FROM PRODUTO;
```

6. Tarefa de consultas compostas

- Consultas propostas na **BD EMPRESA**.

```
--Seleccionamos a BD EMPRESA
USE EMPRESA;

/*** CONSULTA PROPOSTA 1 ***/
--Código do fabricante e identificador daqueles produtos
--con prezo superior a 60€ ou que teñan pedidos de
--cantidade inferior a 5 unidades. Ordenar por
--fabricante e produto.
SELECT cod_fabricante, identificador
FROM PRODUTO
WHERE prezo>60
UNION
SELECT cod_fabricante, id_produto
FROM PEDIDO
WHERE cantidade<5
ORDER BY cod_fabricante, identificador;

/*** CONSULTA PROPOSTA 2 ***/
--Código dos empregados que non fixeron pedidos.
--Ordenar por número de empregado maior.
SELECT numero
FROM EMPREGADO
EXCEPT
SELECT num_empregado
FROM PEDIDO
ORDER BY numero desc;

/*** CONSULTA PROPOSTA 3 ***/
--Código dos clientes que fixeron pedidos e
--con límite de crédito maior ou igual a 40000.
--Usa unha diferenza para resolver esta consulta.
SELECT num_cliente
FROM PEDIDO
EXCEPT
SELECT numero
FROM CLIENTE
WHERE limite_de_credito <40000;
```

```
/*** CONSULTA PROPOSTA 4 ***/
--Código dos clientes que fixeron pedidos e
--con límite de crédito maior ou igual a 40000.
--Usa unha intersección para resolver esta consulta.
--Ordena por código de cliente en orde ascendente.
SELECT num_cliente
FROM PEDIDO
INTERSECT
SELECT numero
FROM CLIENTE
WHERE limite_de_credito >=40000
ORDER BY num_cliente;

/*** CONSULTA PROPOSTA 5 ***/
--Código dos empregados que son directores de sucursal
--ou teñen unha cota de vendas superior a 250000€.
--Solución1
SELECT num_empregado_director
FROM SUCURSAL
UNION
SELECT numero
FROM EMPREGADO
WHERE cota_de_vendas > 250000;
--Solución1
SELECT num_empregado_director
FROM SUCURSAL
UNION ALL
SELECT numero
FROM EMPREGADO
WHERE cota_de_vendas > 250000;
```

7. Tarefa de consultas complexas optimizadas

- Consultas propostas nas BBDD **EMPRESA** e **SOCIEDADE_CULTURAL**:

```
/*** CONSULTA PROPOSTA 1 ***/
--Nif e nome completo nunha columna (ape1 ape2, nome)
--dos socios que deben algunha actividade.
--Nunha segunda columna aparecerá o importe total que debe en actividades.
--Seleccionamos a BD
USE SOCIEDADE_CULTURAL;
SELECT s.nif,
       rtrim(s.ape1+' '+isnull(s.ape2,''))+', '+s.nome as nome_completo,
       sum(a.prezo) as cantidade_debe
FROM SOCIO s, SOCIO_REALIZA_ACTI sr, ACTIVIDADE a
WHERE sr.pagada='N' AND
      s.numero=sr.num_socio AND
        sr.id_actividade=a.identificador
GROUP BY s.numero, s.nif, s.ape1, s.ape2, s.nome
ORDER BY s.ape1, s.ape2, s.nome;
```

```
/*** CONSULTA PROPOSTA 2 ***/
--Número de pedido, descrición e prezo do produto, unidades vendidas e
importe
--de todos os pedidos da BD ordenados de maior a menor importe e
--pola descrición do produto.
--Seleccionamos a BD
USE EMPRESA;
GO
SELECT p.numero, pr.descricion, pr.prezo, p.cantidade,
       p.cantidade*pr.prezo as importe
FROM PEDIDO p INNER JOIN PRODUTO pr
     ON p.cod_fabricante=pr.cod_fabricante AND
          p.id_produto=pr.identificador
ORDER BY importe DESC, pr.descricion;

/*** CONSULTA PROPOSTA 3 ***/
--Número de pedido, descrición e prezo do produto, unidades vendidas e
importe
--de todos os pedidos da BD con importe superior a 1000€,
--ordenados de maior a menor importe e pola descrición do produto.
--Seleccionamos a BD
USE EMPRESA;
GO
SELECT p.numero, pr.descricion, pr.prezo, p.cantidade,
       p.cantidade*pr.prezo as importe
FROM PEDIDO p INNER JOIN PRODUTO pr
     ON p.cod_fabricante=pr.cod_fabricante AND
          p.id_produto=pr.identificador
WHERE p.cantidade*pr.prezo>1000
ORDER BY importe DESC, pr.descricion;

/*** CONSULTA PROPOSTA 4 ***/
--Número de pedido, nome do cliente e data de pedido dos pedidos
--recibidos nos días en que se contrataron empregados.
--No resultado deben aparecer primeiro os pedidos máis recentes.
--Seleccionamos a BD
USE EMPRESA;
GO
SELECT p.numero, c.nome, p.data_pedido
FROM  PEDIDO p INNER JOIN CLIENTE c
        ON p.num_cliente=c.numero
WHERE convert(char(10),p.data_pedido,103) IN (SELECT data_contrato
                                              FROM EMPREGADO)
ORDER BY  p.data_pedido DESC;
```

```
/*** CONSULTA PROPOSTA 5 ***/
--Nome completo dos empregados co nome do empregado que teñen por xefe.
--Seleccionamos a BD
USE EMPRESA;
GO
SELECT rtrim(e.ape1+' '+isnull(e.ape2,''))+', '+e.nome as empregado,
         isnull(rtrim(x.ape1+' '+isnull(x.ape2,''))+', '+x.nome,
                'XEFE POR DETERMINAR') as xefe
FROM EMPREGADO e LEFT JOIN EMPREGADO x
     ON e.num_empregado_xefe=x.numero;

/*** CONSULTA PROPOSTA 6 ***/
--Gasto en actividades por socio.
--Seleccionamos a BD
USE SOCIEDADE_CULTURAL;
GO
SELECT s.nif, sum(a.prezo) as gasto_actividades
FROM SOCIO s, SOCIO_REALIZA_ACTI sr, ACTIVIDADE a
WHERE s.numero=sr.num_socio AND
      sr.id_actividade=a.identificador
GROUP BY s.numero, s.nif;

/*** CONSULTA PROPOSTA 7 ***/
--Nome e apelidos das persoas da sociedade cultural.
--Seleccionamos a BD
USE SOCIEDADE_CULTURAL;
GO
SELECT ape1 as apelido1, ape2 as apelido2, nome as nome_propio,
       'É EMPREGADO' as cargo
FROM EMPREGADO
UNION ALL
SELECT ape1 as apelido1, ape2 as apelido2, nome as nome_propio,
       'NON É EMPREGADO'
FROM SOCIO
ORDER BY ape1, ape2, nome;

/*** CONSULTA PROPOSTA 8 ***/
--Empregando unha consulta composta amosa o identificador
--das sucursais que non teñen empregados traballando nelas.
--Seleccionamos a BD
USE EMPRESA;
GO
SELECT identificador
FROM SUCURSAL
EXCEPT
SELECT id_sucursal_traballa
FROM EMPREGADO;
```

```
/*** CONSULTA PROPOSTA 9 ***/
--Nunha columna nome_abreviado amosa os tres primeiros caracteres
--en minúsculas do primeiro apelido de cada empregado.
--Seleccionamos a BD
USE EMPRESA;
GO
SELECT lower(left(ape1,3)) as nome_abreviado
FROM EMPREGADO;

/*** CONSULTA PROPOSTA 10 ***/
--Nome e apelidos, dos socios que cumpren anos no mes actual.
--Seleccionamos a BD
USE SOCIEDADE_CULTURAL;
GO
SELECT ape1 as apelido1, ape2 as apelido2, nome as nome_propio
FROM SOCIO
WHERE month(data_nac)=month(getdate());
```

TAREFAS DE AUTOAVALIACIÓN

1. Tarefa de consultas simples

- **Consulta 1.1.** BD EMPRESA. Consulta que devolva o código e nome dos fabricantes cuxo nome non ten por segunda letra O.

```
--Solución consulta 1.1
USE EMPRESA;
GO
SELECT codigo, nome
FROM FABRICANTE
WHERE nome NOT LIKE '_O%';
```

- **Consulta 1.2.** BD EMPRESA. Número identificador, nome completo e data de nacemento dos empregados que traballan na sucursal con identificador 12, e naceron no ano 1985. No resultado aparecerán primeiro os empregados máis novos.

```
--Solución consulta 1.2
USE EMPRESA;
GO
SELECT numero, nome, ape1, ape2, data_nacemento
FROM EMPREGADO
WHERE id_sucursal_traballa=12 AND
      data_nacemento BETWEEN '1/1/1985' AND '31/12/1985'
ORDER BY data_nacemento DESC;
```

- **Consulta 1.3.** BD EMPRESA. Número de pedido daqueles nos que se pediron 6 ou 10 unidades. Para resolver esta consulta non se poden utilizar operadores de comparación (>, <, >=, <=, < >, !=).

```
--Solución consulta 1.3
USE EMPRESA;
GO
SELECT numero
FROM PEDIDO
WHERE cantidade IN (6,10);
```

- **Consulta 1.4.** BD EMPRESA. Número e nome propio (nunha única columna separados por un guión, *número - nome_propio*) dos empregados que non teñen segundo apelido. A columna do resultado deberá chamarse *datos_empregados*.

```
--Solución consulta 1.4
USE EMPRESA;
GO
SELECT cast(numero as varchar(6))+' - '+nome as datos_empregados
FROM EMPREGADO
WHERE ape2 is NULL;
```

2. Tarefa de consultas resumo

- **Consulta 2.1.** BD EMPRESA. Cantidade total de empregados que hai por sucursal. Aparecerá o identificador da sucursal e nunha segunda columna a cantidade de empregados que traballan na mesma.

```
--Solución consulta 2.1
USE EMPRESA;
GO
SELECT id_sucursal_traballa, count(*) total_empregados
FROM EMPREGADO
GROUP BY id_sucursal_traballa;
```

- **Consulta 2.2.** BD EMPRESA. Prezo medio dos produtos por fabricante. Nunha primeira columna aparecerá o código de tres caracteres do fabricante, e na segunda o prezo medio dos produtos dese fabricante. No resultado deberán aparecer os fabricante teñan produtos con maior prezo medio.

```
--Solución consulta 2.2
USE EMPRESA;
GO
SELECT cod_fabricante, avg(prezo) as prezo_medio
FROM PRODUTO
GROUP BY cod_fabricante
ORDER BY prezo_medio DESC;
```

- **Consulta 2.3.** BD EMPRESA. Repite a consulta anterior, pero agora só poden aparecer os fabricantes con número de produtos a venda superior a 3.

```
--Solución consulta 2.3
USE EMPRESA;
GO
SELECT cod_fabricante, avg(prezo) as prezo_medio
FROM PRODUTO
GROUP BY cod_fabricante
HAVING count(*) >3
ORDER BY prezo_medio DESC;
```

- **Consulta 2.4.** BD EMPRESA. Cantidade de empregados que son directores dalgunha sucursal. Ten en conta que distintas sucursais poden ter o mesmo director.

```
--Solución consulta 2.4
USE EMPRESA;
GO
SELECT count(DISTINCT num_empregado_director) as total_directores
FROM SUCURSAL;
```

3. Tarefa de consultas con combinacións

- **Consulta 3.1.** BD SOCIEDADE_CULTURAL. Nome das actividades co nome completo do profesor/a que as imparten, só para as actividades que custan máis de 70€. Na primeira columna *Actividade* aparecerá o nome da actividade e na segunda de nome *Docente*, aparecerá o nome completo do docente co formato *apelido1 apelido2, nome.* Deberanse propoñer dúas solucións, unha primeira coa condición de combinación na cláusula FROM, e unha segunda coa condición de combinación na cláusula WHERE.

```
--Solución consulta 3.1
USE SOCIEDADE_CULTURAL;
GO
SELECT a.nome as Actividade,
       rtrim(e.ape1+' '+isnull(e.ape2,''))+','+e.nome as Docente
FROM ACTIVIDADE a INNER JOIN EMPREGADO e
     ON a.num_profesorado_imparte=e.numero
WHERE a.prezo>70;
--Solución consulta 3.1 sen INNER JOIN
SELECT a.nome as Actividade,
       rtrim(e.ape1+' '+isnull(e.ape2,''))+','+e.nome as Docente
FROM ACTIVIDADE a, EMPREGADO e
WHERE a.prezo>70 AND
        a.num_profesorado_imparte=e.numero;
```

- **Consulta 3.2.** BD EMPRESA. Listaxe dos produtos da BD ordenados alfabeticamente por descrición. No resultado aparecerán dúas columnas, na primeira o nome do fabricante e na segunda a descrición do produto. As columnas chamaranse *Fabricante* e *Produto*. Deberanse propoñer dúas solucións, unha primeira coa condición de combinación na cláusula FROM, e unha segunda coa condición de combinación na cláusula WHERE.

```
--Solución consulta 3.2
USE EMPRESA;
GO
--Solución con INNER JOIN
SELECT f.nome as Fabricante, p.descricion as Produto
FROM FABRICANTE f INNER JOIN PRODUTO p
       ON f.codigo=p.cod_fabricante
ORDER BY p.descricion;
--Solución consulta 3.2 sen INNER JOIN
SELECT f.nome as Fabricante, p.descricion as Produto
FROM FABRICANTE f, PRODUTO p
WHERE f.codigo=p.cod_fabricante
ORDER BY p.descricion;
```

- **Consulta 3.3.** BD SOCIEDADE_CULTURAL. Deséxase saber quen paga cada cota, e tamén se hai cotas que non están asignadas a ningún socio ou socios que non teñen cota asignada. A consulta deberá devolver dúas columnas *Socio/a* e *Cota*. En *Socio* aparecerán os números de cada socio e en *Cota* os nomes de cada cota. Se un socio non tivese cota asignada, na columna *Cota* aparecerá a frase -SEN DESIGNAR-. Se unha cota non está asociada a ningún socio, na columna *Socio/a* aparecerá a frase -SEN SOCIO/A-. Para que a consulta teña sentido débese supoñer que a columna *cod_cota* de SOCIO puidese admitir nulos.

```
--Solución consulta 3.3
USE SOCIEDADE_CULTURAL;
GO
SELECT isnull(cast(s.numero as varchar(11)),'-SEN SOCIO/A-') as "Socio/a",
       isnull(c.nome,'-SEN DESIGNAR-') as Cota
FROM SOCIO s FULL JOIN COTA c
     ON s.cod_cota=c.codigo;
```

- **Consulta 3.4.** BD EMPRESA. Consulta que devolva cada empregado co cliente ou clientes que ten asignados. Se un empregado ten máis dun cliente aparecerá en tantas filas como clientes teña. Na primeira columna *Empregado* aparecerá o número e o primeiro apelido do empregado separados por un guión. Na segunda columna *Cliente* aparecerá o nome do mesmo. Se un empregado non ten clientes especificarase -SEN CLIENTES- e se un cliente non ten empregado indicarase deixando en branco o empregado. Na solución na cláusula FROM a primeira táboa que se debe poñer será a de EMPREGADO.

```
--Solución consulta 3.4
USE EMPRESA;
GO
SELECT isnull(cast(e.numero as varchar(6))+' - '+e.ape1, '') as Empregado,
       isnull(c.nome,'-SEN CLIENTES-') as Cliente
FROM EMPREGADO e LEFT JOIN CLIENTE c
     ON e.numero=c.num_empregado_asignado;
```

- **Consulta 3.5.** BD SOCIEDADE_CULTURAL. Empregados coas actividades que imparten. No resultado aparecerá na primeira columna *Empregado* o número da seguridade social (NSS) do empregado, e na segunda columna *Actividades* o nome da actividade. Se un empregado non imparte actividades aparecerá a frase -SEN ACTIVIDADES- e se imparte varias aparecerá en varias filas, cada unha cunha das actividades. Na solución na cláusula FROM a primeira táboa que se debe poñer será a de ACTIVIDADE.

```
--Solución consulta 3.5
USE SOCIEDADE_CULTURAL;
GO
SELECT e.nss as Empregado,
       isnull(a.nome,'-SEN ACTIVIDADES-') as Actividades
FROM ACTIVIDADE a RIGHT JOIN EMPREGADO e
     ON a.num_profesorado_imparte=e.numero;
```

- **Consulta 3.6.** BD SOCIEDADE_CULTURAL. Produto cartesiano entre a táboa de aulas e a de actividades. Nunha primeira columna aparecerá o nome da aula e na segunda o nome da actividade. Débense propoñer dúas solucións, segundo as sintaxes estudadas para o produto cartesiano.

```
--Solución consulta 3.6
USE SOCIEDADE_CULTURAL;
GO
--Solución con CROSS JOIN
SELECT au.descricion, ac.nome
FROM AULA au CROSS JOIN ACTIVIDADE ac;
--Solución sen CROSS JOIN
SELECT au.descricion, ac.nome
FROM AULA au, ACTIVIDADE ac;
```

- **Consulta 3.7.** BD SOCIEDADE_CULTURAL. Consulta que devolva o NIF de cada socio combinado con cada un dos nomes das cotas da BD. O resultado terá unha columna co formato *NIF socio - Nome cota*. Débense propoñer dúas solucións, segundo cómo se dispoñan as táboas no FROM.

```
--Solución consulta 3.7
USE SOCIEDADE_CULTURAL;
GO
--Solución con CROSS JOIN
SELECT s.nif+' - '+c.nome as Socio_Cota
FROM SOCIO s CROSS JOIN COTA c;
--Solución sen CROSS JOIN
SELECT s.nif+' - '+c.nome as Socio_Cota
FROM SOCIO s, COTA c;
```

4. Tarefa de consultas con subconsultas

- **Consulta 4.1.** BD EMPRESA. Número e data dos pedidos realizados por empregados sen cota de vendas. Deberase resolver sen usar combinacións de táboas.

```
--Solución consulta 4.1
USE EMPRESA;
GO
SELECT numero, data_pedido
FROM PEDIDO
WHERE num_empregado IN (SELECT numero
                        FROM EMPREGADO
                        WHERE cota_de_vendas IS NULL);
```

- **Consulta 4.2.** BD SOCIEDADE_CULTURAL. Nome e prezo das actividades con prezo superior ao prezo medio das cotas. Deberase resolver sen usar combinacións de táboas.

```
--Solución consulta 4.2
USE SOCIEDADE_CULTURAL;
GO
SELECT nome, prezo
FROM ACTIVIDADE
WHERE prezo > (SELECT avg(importe)
               FROM COTA);
```

- **Consulta 4.3.** BD SOCIEDADE_CULTURAL. Listaxe do nif de todos os socios sempre que non exista ningún vivindo na provincia de Madrid. Deberase resolver sen usar combinacións de táboas.

```
--Solución consulta 4.3
USE SOCIEDADE_CULTURAL;
GO
SELECT nif
FROM SOCIO
WHERE NOT EXISTS (SELECT numero
                  FROM SOCIO
                  WHERE cod_provincia_enderezo=(SELECT codigo
                                                FROM PROVINCIA
                                                WHERE nome='Madrid'));
```

- **Consulta 4.4.** BD SOCIEDADE_CULTURAL. Nome e prezo das actividades con prezo superior ao da cota máis cara. Deberase resolver sen usar combinacións de táboas nin a función colectiva max.

```
--Solución consulta 4.4
USE SOCIEDADE_CULTURAL;
GO
SELECT nome, prezo
FROM ACTIVIDADE
WHERE prezo >ALL(SELECT importe
                 FROM COTA);
```

5. Tarefa de consultas con funcións integradas no xestor

- **Consulta 5.1.** BD EMPRESA. Consulta que devolva información dos empregados nas seguintes columnas:
  - número identificador e nome propio separados polo símbolo # (cancelo).
  - Tres primeiras letras do primeiro apelido.
  - Tres últimas letras do primeiro apelido.
  - Terceiro e cuarto caracteres do primeiro apelido.
  - Nome propio en minúsculas.
  - Nome propio en maiúsculas.
  - Título eliminándolle os posibles espazos en branco á esquerda.
  - Título eliminándolle os posibles espazos en branco á dereita.
  - Título eliminándolle todos os espazos.
  - Nome propio substituíndo E por O.

```
--Solución consulta 5.1
USE EMPRESA;
GO
SELECT cast(numero as varchar(6))+' # '+nome as Empregado,
       left(ape1, 3) as Letras_ini,
         right(ape1, 3) as Letras_fin,
         substring(ape1,3,2) as Subcadea_ape1,
         lower(nome) as Nome_minusculas,
         upper(nome) as Nome_maiusculas,
         ltrim(titulo) as Titulo_sen_espazos_ini,
         rtrim(titulo) as Titulo_sen_espazos_fin,
         replace(titulo,' ','') as Titulo_sen_espazos,
         replace(nome,'E','O') as Nome_con_O
FROM EMPREGADO;
```

- **Consulta 5.2.** BD EMPRESA. Consulta que devolva información dos empregados nas seguintes columnas:
  - Nome completo dos empregados co formato *apelido1 apelido2, nome*.
  - Cota de vendas.
  - Cota de vendas aproximada por exceso.
  - Cota de vendas aproximada por defecto.
  - Cota de vendas elevada ao cadrado.
  - Cota de vendas elevada ao cubo.
  - Raíz cadrada da cota de vendas.

  Todos os valores nulos do resultado deberán aparecer substituídos polo valor 0.

```
--Solución consulta 5.2
USE EMPRESA;
GO
SELECT rtrim(ape1+' '+isnull(ape2,''))+', '+nome as nome_completo,
       isnull(cota_de_vendas,0) as cota_de_vendas,
         isnull(ceiling(cota_de_vendas),0) as cota_por_exceso,
         isnull(floor(cota_de_vendas),0) as cota_por_defecto,
         isnull(square(cota_de_vendas),0) as cadrado_cota_vendas,
         isnull(power(cota_de_vendas,3),0) as cubo_cota_vendas,
         isnull(sqrt(cota_de_vendas),0) as raiz_cadrada_cota
FROM EMPREGADO;
```

- **Consulta 5.3.** BD EMPRESA. Consulta que devolva información dos pedidos nas seguintes columnas:
  - Número do pedido.
  - Data actual con axuste de zona horaria.
  - Data do pedido co formato dd-mm-aaaa.
  - Día no que se fixo o pedido.
  - Nome do mes no que se fixo o pedido.
  - Ano no que se fixo o pedido.
  - Meses que pasaron desde que se fixo o pedido.
  - Data de pedido adiantada 4 anos.
  - Data de pedido retrasada 1 mes.

```
--Solución consulta 5.3
USE EMPRESA;
GO
SELECT numero,
       sysdatetimeoffset() as data_con_axuste,
         convert(char(10), data_pedido,105) as data_pedido,
         datepart(day, data_pedido) as dia,
         --day(data_pedido) tamén é válido
         datename(month,data_pedido) as nome_mes,
         year(data_pedido) as ano,
```

```
            --datepart(year, data_pedido)  tamén é válido
            datediff(month, data_pedido, getdate()) as meses_pasados,
            dateadd(year, -4, data_pedido) as data_adiantada_4anos,
            dateadd(month, 1, data_pedido) as data_retrasada_1mes
FROM PEDIDO;
```

- **Consulta 5.4.** BD SOCIEDADE_CULTURAL. Consulta que devolve unha columna *Linguaxe_usada* na que dependendo da linguaxe en uso na sesión, aparecerá unha frase ou outra:
  - se a linguaxe é *Español* deberá aparecer a frase SU IDIOMA ES ESPAÑOL.
  - se a linguaxe é *us_english* deberá aparecer a frase YOUR LANGUAGE IS ENGLISH.
  - se a linguaxe é outra deberá aparecer a frase VOSTEDE NON ESTÁ USANDO NIN ESPAÑOL NIN INGLÉS.

```
--Solución consulta 5.4
USE SOCIEDADE_CULTURAL;
GO
SELECT CASE
            WHEN @@language='Español' THEN 'SU IDIOMA ES ESPAÑOL'
            WHEN @@language='us_english' THEN 'YOUR LANGUAGE IS ENGLISH'
            ELSE 'VOSTEDE NON ESTÁ USANDO NIN ESPAÑOL NIN INGLÉS'
       END  as Linguaxe_usada;
```

6. Tarefa de consultas compostas

- **Consulta 6.1.** BD SOCIEDADE_CULTURAL. Empregando unha consulta composta pídese a listaxe dos identificadores das actividades con prezo superior a 15€ ou que se impartan en aulas con superficie inferior a 100 metros cadrados. O resultado aparecerá ordenado de maior a menor identificador.

```
--Solución consulta 6.1
USE SOCIEDADE_CULTURAL;
GO
SELECT identificador
FROM  ACTIVIDADE
WHERE prezo>15
UNION
SELECT ac.identificador
FROM ACTIVIDADE ac INNER JOIN AULA au
     ON ac.num_aula=au.numero
WHERE au.superficie<100
ORDER BY identificador DESC;
```

- **Consulta 6.2.** BD SOCIEDADE_CULTURAL. Empregando unha consulta composta amosar o nif dos socios que asisten a actividades e que pagaron a cota anual.

```
--Solución consulta 6.2
USE SOCIEDADE_CULTURAL;
GO
SELECT s.nif
FROM SOCIO s INNER JOIN SOCIO_REALIZA_ACTI sr
     ON s.numero=sr.num_socio
INTERSECT
SELECT nif
FROM SOCIO
WHERE abonada='S';
```

- **Consulta 6.3.** BD EMPRESA. Empregando unha consulta composta amosar o número identificador dos clientes que aínda non fixeron pedidos. Ordena o resultado polo número de cliente en orde ascendente.

```
--Solución consulta 6.3
USE EMPRESA;
GO
SELECT numero
FROM CLIENTE
EXCEPT
SELECT num_cliente
FROM PEDIDO
ORDER BY numero;
```

7. Tarefa de consultas complexas optimizadas

- **Consulta 7.1.** Consulta que devolve na primeira columna a descrición dos produtos, e nunha segunda columna o gasto total en pedidos dese produto. No resultado só poderán aparecer aqueles produtos cuxo gasto medio é menor de 1000€.

  A primeira columna chamarase *Produto* e a segunda *Gasto total*.

  Se de algún produto non se realizaron pedidos na columna do Gasto total deberá aparecer o número cero.

  Deberán aparecer primeiro no resultado os produtos con maior gasto total.

```
--Solución consulta 7.1
USE EMPRESA;
GO
SELECT  p.descricion as Produto,
       convert(numeric(8,2),isnull(sum(p.prezo*pe.cantidade),0)) as "Gasto total"
FROM PRODUTO p LEFT JOIN PEDIDO pe on p.cod_fabricante=pe.cod_fabricante
       AND     p.identificador=pe.id_produto
GROUP BY p.cod_fabricante, p.identificador, p.descricion, p.prezo
HAVING isnull(sum(p.prezo*pe.cantidade),0)<1000
ORDER BY "Gasto total" DESC;
```

- **Consulta 7.2.** Consulta que devolva a lista dos pedidos que hai máis de 12 meses e menos de 25 meses que se realizaron.

  Para facer a comprobación dos anos que hai que se realizou o pedido non está permitido usar nin o predicado IN, nin OR nin tampouco os operadores de comparación >=, >, <=, !=, = <>.

  No resultado deberá aparecer o número do pedido, nunha segunda columna *FechaPed* aparecerá a data do pedido con formato dd-mm-aaaa (separación empregando guións), e nunha terceira columna de nome *Unidades* aparecerá o seguinte en función do número de unidades solicitadas no pedido:

  - Se a cantidade de unidades do pedido é inferior a 15 aparecerá o texto POUCAS.
  - Se a cantidade de unidades do pedido é superior ou igual a 15 e menor que 20 aparecerá o texto NORMAL.
  - Se a cantidade de unidades do pedido é superior ou igual a 20 aparecerá o texto MOITAS.

  Deben aparecer os pedidos máis recentes primeiro.

```
--Solución consulta 7.2
USE EMPRESA;
GO
SELECT  numero, convert(char(10),data_pedido,105) as FechaPed,
            CASE
                  WHEN cantidade<15 THEN 'POUCAS'
                  WHEN cantidade BETWEEN 15 AND 19 THEN 'NORMAL'
                  ELSE 'MOITAS'
            END as Unidades
FROM PEDIDO
WHERE datediff(month,data_pedido,getdate()) BETWEEN 12 AND 25
ORDER BY FechaPed DESC;
```

- **Consulta 7.3.** Consulta que devolva a cidade en maiúsculas de cada unha das sucursais da BD, a súa rexión en minúsculas e noutra terceira columna de nome *pedido_minimo* as unidades do pedido con menor número de unidades da sucursal. Ten en conta que un pedido pertence a unha sucursal se foi realizado por un empregado que traballa nesa sucursal.

  Se existise algunha sucursal que non teña vendido nada, non aparecerá no resultado.

```
--Solución consulta 7.3
USE EMPRESA;
GO
SELECT upper(s.cidade) as cidade, lower(s.rexion) as rexion,
       min(p.cantidade) as pedido_minimo
FROM SUCURSAL s, EMPREGADO e, PEDIDO p
WHERE s.identificador=e.id_sucursal_traballa AND
        e.numero=p.num_empregado
GROUP BY s.identificador, s.cidade, s.rexion;
```

- **Consulta 7.4.** Nome completo nunha columna (co formato *ape1 ape2, nome*) e noutra a data de contratación (co formato dd/mm/aaaa), dos empregados que foron contratados un día 12, 19 ou 20 de calquera mes e de calquera ano. Importante: Non se pode usar o operador OR nin a función *day* para resolver a consulta.

```
--Solución consulta 7.4
USE EMPRESA;
GO
SELECT rtrim(ape1+' '+isnull(ape2,''))+', '+nome as nome_completo,
       convert(char(10),data_contrato, 105) as data_contrato
FROM EMPREGADO
WHERE datepart(dd,data_contrato) IN(12, 19,20);
```

- **Consulta 7.5.** Executa no servidor a seguinte instrución e comproba que foi agregado un produto novo coa descrición PROBA_AVALIACION.

```
INSERT INTO PRODUTO
(cod_fabricante, identificador, descricion, prezo, existencias)
VALUES ('ASU', '41777', 'PROBA_AVALIACION', 300, 40);
```

Unha vez comprobado que a nova fila está na táboa, realiza a consulta que devolve o número total de produtos cuxa descrición contén algún dos seguintes tres caracteres: / (barra diagonal ou slash), _ (guión baixo), - (guión medio). Para facelo non está permitido o uso dos corchetes ([ ]).

/ (barra diagonal ou slash)	_ (guión baixo)	- (guión medio)

Unha vez rematada a consulta é necesario eliminar o produto engadido coa seguinte instrución:

```
DELETE FROM PRODUTO WHERE descricion='PROBA_AVALIACION';
```

```
--Solución consulta 7.5
USE EMPRESA;
GO
INSERT INTO PRODUTO
(cod_fabricante,identificador,descricion,prezo,existencias)
VALUES ('ASU', '41777', 'PROBA_AVALIACION', 300, 40);

SELECT count(identificador) as produtos_cons_simbolos_especiais
FROM PRODUTO
WHERE descricion LIKE '%\/%' ESCAPE '\' OR
        descricion LIKE '%\_%' ESCAPE '\' OR
        descricion LIKE '%-%';

DELETE FROM PRODUTO WHERE descricion='PROBA_AVALIACION';
```

- **Consulta 7.6.** Número, data e cantidade, dos pedidos con maior número de unidades. No resultado aparecerán primeiro os pedidos máis antigos.

```
--Solución consulta 7.6
USE EMPRESA;
GO
SELECT numero, data_pedido, cantidade
FROM PEDIDO
WHERE cantidade=(SELECT max(cantidade)
                          FROM PEDIDO)
ORDER BY data_pedido;
```

- **Consulta 7.7.** Consulta que devolva o 25% dos empregados cuxo primeiro apelido ten por segunda letra un A. A consulta amosará en diferentes columnas o número, nome propio, o primeiro apelido dos empregados, o número de caracteres do primeiro apelido e o nome do mes en que foi contratado. Para resolver a busca da letra A na segunda posición do primeiro apelido, deberase empregar unha función integrada no xestor.

```
--Solución consulta 7.7
USE EMPRESA;
GO
SELECT TOP 25 PERCENT numero, nome, ape1, len(ape1) as lonx_ape1,
          datename(month,data_contrato) as nome_mes_contrato
FROM EMPREGADO
WHERE substring(ape1,2,1)='A';
```

- **Consulta 7.8.** Descrición dos produtos dos que non se fixeron pedidos. Para obter o resultado non se pode empregar ningunha combinación externa, nin a palabra reservada DISTINCT, nin subconsultas.

```
--Solución consulta 7.8
USE EMPRESA;
GO
SELECT descricion
FROM PRODUTO
EXCEPT
SELECT pr.descricion
FROM PRODUTO pr, PEDIDO p
WHERE pr.cod_fabricante=p.cod_fabricante AND
              pr.identificador=p.id_produto;
```

- **Consulta 7.9.** Nunha consulta haberá que devolver a hora actual (só a hora, sen minutos), as data e hora actuais con axuste de zona horaria (seguindo o estándar *Europeo predeterminado+milisegundos*), o nome da linguaxe do servidor e o número máximo de conexións permitidas no servidor.

```
--Solución consulta 7.9
USE MASTER;
GO
SELECT datepart(hh,getdate()) as Hora_actual,
         convert(char(34),sysdatetimeoffset(),113) as DataHoraConAxuste,
         @@LANGUAGE as Ling_server,
         @@MAX_CONNECTIONS as Conexions_maximas;
```

- **Consulta 7.10.** Nome de cada cliente e ao lado do mesmo as unidades vendidas de media (columna *cantidade*) nos seus pedidos, só para aqueles clientes cuxo representante de vendas (empregado que ten asociado) ten por título RP VENDAS. Se non mercaron nada deberá aparecer como media de unidades 0.

  A columna das unidades medias deberá ter 3 díxitos como máximo na parte enteira e 2 decimais.

  Deberán aparecer primeiro no resultado os clientes con maior número de unidades medias.

```
--Solución consulta 7.10
USE EMPRESA;
GO
SELECT c.nome,
       convert(numeric(5,2),isnull(avg(p.cantidade),0)) as media_unidades
FROM (CLIENTE c LEFT JOIN PEDIDO p
         ON c.numero=p.num_cliente) INNER JOIN EMPREGADO e
         ON c.num_empregado_asignado=e.numero
WHERE e.titulo='RP VENDAS'
GROUP BY c.numero, c.nome
ORDER BY media_unidades DESC;
```

- **Consulta 7.11.** Produto cartesiano de FABRICANTE e PRODUTO. No resultado aparecerán dúas columnas, o nome do fabricante e a descrición do produto. Esta consulta deberá facerse sen empregar no FROM combinacións internas.

```
--Solución consulta 7.11
USE EMPRESA;
GO
SELECT f.nome, p.descricion
FROM FABRICANTE f CROSS JOIN PRODUTO p;
```

- **Consulta 7.12.** Produto cartesiano de FABRICANTE e PRODUTO. No resultado aparecerán dúas columnas, o nome do fabricante e a descrición do produto. Esta consulta deberá facerse empregando no FROM só combinacións internas.

```
--Solución consulta 7.12
USE EMPRESA;
GO
SELECT f.nome, p.descricion
FROM FABRICANTE f, PRODUTO p;
```

- **Consulta 7.13.** Nome dos fabricantes dos que non hai produtos na BD.

```
--Solucións consulta 7.13
USE EMPRESA;
GO
--Solución1
SELECT nome
FROM FABRICANTE
WHERE NOT EXISTS (SELECT identificador
                  FROM PRODUTO
                           WHERE cod_fabricante=codigo);
--Solución2
SELECT f.nome
FROM FABRICANTE f LEFT JOIN PRODUTO p
     ON f.codigo=p.cod_fabricante
WHERE p.identificador IS NULL;
--Solución3 (NON OPTIMIZADA)
SELECT nome
FROM FABRICANTE
WHERE codigo NOT IN (SELECT cod_fabricante
                     FROM PRODUTO);
```

- **Consulta 7.14.** Número identificador e data dos pedidos que teñan 9,7,10 ou 8 unidades e que foron comprados polo cliente 1108 ou vendidos polo empregado 101.

```
--Solución consulta 7.14
USE EMPRESA;
GO
SELECT numero, data_pedido
FROM PEDIDO
WHERE (num_cliente=1108 OR
       num_empregado=101)  AND
      cantidade BETWEEN 7 AND 10;
```

- **Consulta 7.15.** Nomes de tódolos clientes e de tódolos fabricantes da BD nunha única columna. A columna que se chamará *empresas*, deberá aparecer ordenada alfabeticamente.

```
--Solución consulta 7.15
USE EMPRESA;
GO
SELECT nome as empresas FROM CLIENTE
UNION ALL
SELECT nome  FROM FABRICANTE
ORDER BY nome;
```