

Autor: Pablo Palanques Gil

Voluntariament, a continuació estan tots els exercicis propostos del Tema 6:

En la BD **factura**, connectant com a usuari **factura**:

6.1 Traure tota la informació dels pobles (anomeneu-la **Ex_6_1.sql**).

```
select * from poble;
```

6.2 Traure el codi postal, el nom i l'adreça, per aquest ordre, de tots els venedors (anomeneu-la **Ex_6_2.sql**).

```
select cp, nom, adreca from venedor;
```

6.3 Traure el codi d'article, la descripció, preu i preu incrementat en un 5%, de tots els articles.

```
select cod_a, descrip, preu, trunc(preu*1.05, 2) from article;
```

6.4 Traure la informació dels clients amb el següent format (ha d'anar tot en una columna):

Damborenea Corbato, Alicia. CALLE MADRID, 83 (12425)

Fixeu-vos que està tot en una columna, i per tant haureu de concatenar de la forma adequada. Fixeu-vos també que en el nom només les inicials estan en majúscules

```
select INITCAP(nom) || '. ' || adreca || ' (' || cp || ')' from client;
```

6.5 Traure el num_f, data i cod_ven de les factures amb les següents capçaleres respectivament: **Número Factura**, **data** i **Codi Venedor** (anomeneu-lo **Ex_6_5**)

```
select num_f as "Número Factura", data, cod_ven as "Codi Venedor" from factura;
```

6.6 Donar àlies als camps que ho necessiten de la taula ARTICLE (anomeneu-lo **Ex_6_6**)

```
select cod_a as "Codi Article", descrip as "Descripció", preu, stock, stock_min as "Stock Mínim", cod_cat as "Categoria" from article;
```

6.7 Traure els **clients** de la **ciutat** amb codi **12309**.

```
select * from client where cod_pob = 12309;
```

6.8 Traure totes les **factures** del mes de **març** de **2015**.

```
select * from factura where data BETWEEN '2015-03-01' and '2015-03-31';
```

6.9 Traure tots els articles de la categoria **BjcOlimpia** amb un **stock** entre 2 i 7 unitats.

```
select * from article where cod_cat = 'BjcOlimpia' and stock between 2 and 7;
```

6.10 Traure tots els **clients** que **no** tenen introduït el **codi postal**.

```
select * from client where cp is null;
```

6.11 Traure tots els **articles** amb el **stock** introduït però que **no** tenen introduït el **stock mínim**.

```
select * from article where stock is not null and stock_min is null;
```

6.12 Traure tots els **clients**, el **primer cognom** dels quals és **VILLALONGA**.

```
select * from client where nom like 'VILLALONGA %';
```

6.13.a Modificar l'anterior per a traure tots els que són **VILLALONGA** de **primer** o de **segon** cognom.

```
select * from client where nom like 'VILLALONGA %' or nom like '%VILLALONGA,%';
```

6.13.b Modificar l'anterior per a traure tots els que **no** són **VILLALONGA** ni de primer ni de segon cognom.

```
select * from client where nom not like 'VILLALONGA %' and nom not like '%VILLALONGA,%';
```

6.14 Traure els **articles** "**Pulsador**" (la descripció conté aquesta paraula), el **preu** dels quals oscila entre 2 i 4 € i dels quals tenim un **stock** estrictament **major** que el **stock mínim**.

```
select * from article where descrip like '%Pulsador%' and preu between 2 and 4 and stock > stock_min;
```

6.15 Comptar el nombre de **clients** que tenen el **codi postal nul**.

```
select count(*) from client where cp is null;
```

6.16 Comptar el número de vegades que l'article **L76104** entra en les línies de factura, i el número total d'unitats venudes d'aquest article. Només us fa falta la taula **LINIA_FAC**.

```
select count(*), sum(quant) from linia_fac lf where cod_a = 'L76104';
```

6.17 Traure la **mitjana** del **stock** dels articles.

```
select AVG(stock) from article;
```

6.18 Modificar l'anterior per a **tenir en compte els valors nuls, com si foren 0**. Us vindrà bé la funció **COALESCE** que converteix els nuls del primer paràmetre al valor donat com a segon paràmetre (si és diferent de nul, deixa igual el valor). Per tant l'heu d'utilitzar d'aquesta manera: **COALESCE(stock,0)**

```
select AVG(coalesce(stock, 0)) from article;
```

6.19 Comptar **quantes factures té el client 375**

```
select count(*) from factura where cod_cli = 375;
```

6.20 Calcular el **descompte màxim**, el **mínim** i el descompte **mitjà** de les **factures**.

```
select MAX(dte), MIN(dte), AVG(dte) from factura;
```

6.21 Comptar el número de pobles de cada província (és suficient traure el codi de la província i el número de pobles).

```
select cod_pro, count(*) from poble group by cod_pro;
```

6.22 Comptar el nombre de clients en cada poble i codi postal.

```
select cod_pob, cp, count(*) from client group by cod_pob, cp;
```

6.23 Comptar el número de factures de cada venedor a cada client.

```
select cod_ven, cod_cli, count(*) from factura group by cod_ven, cod_cli;
```

6.24 Comptar el número de factures de cada trimestre. Per a poder traure el trimestre i agrupar per ell (ens val el número de trimestre, que va del 1 al 4), podem utilitzar la funció **TO_CHAR(data,'Q')**.

```
select TO_CHAR(data, 'Q'), count(*) from factura group by TO_CHAR(data, 'Q');
```

6.25 Calcular quantes vegades s'ha venut un article, la suma d'unitats venudes, la quantitat màxima i la quantitat mínima.

```
select cod_a, count(*), SUM(quant), MAX(quant), MIN(quant) from linia_fac group by cod_a;
```

6.26 Comptar el número d'articles de cada categoria i el preu mitjà.

```
select cod_cat, count(*), AVG(preu) from article group by cod_cat;
```

6.27 Calcular el total de cada factura, sense aplicar descomptes ni IVA. Només ens farà falta la taula **LINIES_FAC**, i consistirà en agrupar per cada **num_f** per a calcular la suma del **preu** multiplicat per la **quantitat**.

```
select num_f, SUM(quant*preu) from linia_fac group by num_f;
```

6.28 Calcular la mitjana de quantitats demanades d'aquells articles que s'han demanat més de dues vegades. Observeu que la taula que ens fa falta és LINIA_FAC, i que la condició (en el HAVING) és sobre el número de vegades que entra l'article en una línia de factura, però el resultat que s'ha de mostrar és la mitjana de la quantitat.

```
select cod_a, count(quant), AVG(quant) from linia_fac group by cod_a
having count(quant) > 2;
```

6.29 Traure els pobles que tenen entre 3 i 7 clients. Traure només el codi del poble i aquest número

```
select count(cod_cli), cod_pob from client group by cod_pob having
count(cod_cli) between 3 and 7;
```

6.30 Traure les categories que tenen més d'un article "car" (de més de 100 €). Observeu que també ens eixirà la categoria NULL, és a dir, apareixerà com una categoria aquells articles que no estan catalogats.

```
select cod_cat, count(preu) from article where preu > 100 group by
cod_cat having count(cod_a) > 1;
```

6.31 Traure els clients que tenen més d'una factura, amb el número de factures.

```
select cod_cli, count(num_f) from factura group by cod_cli having
count(num_f) > 1;
```

6.32 Modificar l'anterior per a traure els clients que tenen més d'una factura en el primer trimestre.

```
select cod_cli, count(num_f) from factura where TO_CHAR(data,'Q') like
'1' group by cod_cli having count(num_f) > 1;
```

6.33 Calcular el total de cada factura d'aquelles factures que tenen 10 o més línies de factura, sense aplicar descomptes ni IVA (com la consulta **6.26**), i també aplicant el descompte que consta en la línia de factura (no el descompte de tota la factura). Tindrem el problema que el valor NULL és especial, i en operar amb qualsevol altre valor donarà NULL. En aquest cas clarament l'hem de considerar com un descompte 0. Podeu utilitzar una funció que substitueix els valors nuls trobats en el primer paràmetre, pel segon paràmetre d'aquesta manera: **COALESCE(dte,0)**

```
select num_f, SUM(quant*preu), SUM(quant*preu*(1-
0.01*coalesce(dte,0))) from linia_fac group by num_f having
count(num_l)>=10;
```

6.34 Traure tots els clients ordenats per codi de població, i dins d'aquests per codi postal.

```
select * from client order by cod_pob, cp;
```

6.35 Traure tots els articles ordenats per la categoria, dins d'aquest pel stock, i dins d'aquest per preu (de forma descendent)

```
select * from article order by cod_cat, stock, preu desc;
```

6.36 Traure els resultats de la consulta **6.33** ordenats pel total de la factura quan ja s'ha aplicat el descompte, de forma descendent.

```
select num_f, SUM(quant*preu), SUM(quant*preu*(1-  
0.01*coalesce(dte,0))) from linia_fac group by num_f having  
count(num_1)>=10 order by 3 DESC;
```

6.37 Traure tots els articles ordenats per la diferència entre el stock i el stock mínim de forma descendent. Com que en moltes ocasions el stock o el stock mínim és nul, hem de considerar en aquestos casos com 0. Per tant hem de tornar a utilitzar la funció **COALESCE(stock,0)** (i també per al stock mínim).

```
select *, COALESCE(stock,0) - COALESCE(stock_min,0) from article order  
by COALESCE(stock,0) - COALESCE(stock_min,0) desc;
```

6.38 Traure els codis de venedor amb el número de factures venudes en el segon semestre de 2015, ordenades per aquest número de forma descendent

```
select cod_ven, count(num_f) from factura where to_char(data,'Q') like  
'2' and to_char(data,'YYYY') like '2015' group by cod_ven order by 2  
desc;
```

6.39 Traure els venedors que han venut alguna cosa el mes de gener de 2015.

```
select distinct cod_ven from factura where to_char(data,'YYYY-MM')  
like '2015-01';
```

6.40 Traure els diferents tipus d'IVA que s'han aplicat a les factures de cada venedor, també durant el mes de gener de 2015

```
select distinct cod_ven, iva from factura where to_char(data,'YYYY-MM')  
like '2015-01';
```

6.41 Traure els diferents caps de venedors (eviteu que aparega el valor nul)

```
select distinct cod_cap from venedor where cod_cap is not null;
```

6.42 Traure els diferents descomptes que s'han aplicat als articles, el codi dels quals comença per **SAT**. Traure tant el codi d'article com el descompte.

```
select distinct cod_a, dte from linia_fac where cod_a like 'SAT%';
```

6.43 Comptar en quantes poblacions tenim clients

```
select count(distinct cod_pob) from client;
```

6.44 Traure tota la informació dels dos articles més cars.

```
select * from article order by preu desc limit 2;
```

6.45 Traure el codi de les tres ciutats amb més clients

```
select cod_pob, count(cod_cli) from client group by cod_pob order by 2 desc limit 3;
```

6.46 Traure el venedor que ha venut menys factures

```
select cod_ven, count(num_f) from factura where cod_ven is not null group by cod_ven order by count(num_f) limit 1;
```

6.47 Traure les tres factures més cares (sense comptar els descomptes)

```
select num_f, SUM(preu*quant) from linia_fac group by num_f order by 2 desc limit 3;
```

6.48 Modificar l'anterior per a traure totes les factures, excepte les 3 més cares.

```
select num_f, SUM(preu*quant) from linia_fac group by num_f order by 2 desc offset 3;
```

6.49 Crear una taula anomenada **ARTICLE_999x**, on 999 han de ser les 3 últimes xifres del DNI, i x la lletra del teu NIF, que siga una còpia de la taula ARTICLE, però substituint els valors nuls de **stock** i **stock_min** per zeros.

```
select cod_a, descrip, preu, COALESCE(stock,0) as stock, COALESCE(stock_min,0) as stock_min, cod_cat into ARTICLE_554z from article;
```

6.50 Utilitzar la taula anterior per a traure el stock màxim, el mínim i la mitjana de stocks. Observeu que si utilitzàrem la taula ARTICLE, els resultats no serien els mateixos (excepte el màxim), sobretot la mitjana, ja que els valors nuls no entrarien en els càlculs d'aquesta mitjana.

```
select MAX(stock), MIN(stock), AVG(stock) from article_554z;
```

CONSULTAS AVANZADAS (SELECT)

6.51 Traure el nom dels clients amb el número de factura i la data (individuals, sense agrupar res) que té cada client. Trau el resultat ordenat per client, i dins d'aquest per data de la factura.

```
select NOM, NUM_F, data from CLIENT inner JOIN FACTURA on client.cod_cli = factura.cod_cli order by 1, 3
```

o bien:

```
select nom, num_f, data
from CLIENT, factura
where client.cod_cli = factura.cod_cli
order by 1, 3
```

6.52 Traure el nom del soci, amb el codi i la descripció de cada article que ha demanat. Ordena per nom del soci i codi de l'article.

```
select nom, linia_fac.cod_a, descrip
from ((client inner join factura on client.cod_cli = factura.cod_cli)
inner join linia_fac on factura.num_f = linia_fac.num_f )
inner join article on linia_fac.cod_a = article.cod_a
order by 1, 2
```

o bien:

```
select nom, linia_fac.cod_a, descrip
from client, factura, linia_fac, article
where client.cod_cli = factura.cod_cli
AND factura.num_f = linia_fac.num_f
AND linia_fac.cod_a = article.cod_a
order by 1, 2
```

6.53 Modificar l'anterior per a que no es repetesquen els resultats.

```
select distinct nom, linia_fac.cod_a, descrip
from client, factura, linia_fac, article
where client.cod_cli = factura.cod_cli
AND factura.num_f = linia_fac.num_f
AND linia_fac.cod_a = article.cod_a
order by 1, 2
```

6.54 Traure el nom dels clients amb la quantitat de factures que tenen, ordenades per aquest número de major a menor

```
select nom, count(factura.num_f) as "numFactures"
from client, factura
where client.cod_cli = factura.cod_cli
group by nom
order by 2 desc
```

6.55 Traure el número de factura, data, codi de client, total de la factura (amb l'àlies IMPORT) i total de la factura aplicant descomptes d'article (amb àlies DESCOMPTE_1), com en la consulta **6.33**, però sense el límit de les 10 línies de factura. Ordena per número de factura.

```
select linia_fac.num_f, data, cod_cli, SUM(quant*preu) as "IMPORT",
SUM(quant*preu*(1-0.01*coalesce(linia_fac.dte,0))) AS "DESCOMPTE_1"
from factura, linia_fac
where factura.num_f = linia_fac.num_f
group by linia_fac.num_f, data, cod_cli
order by linia_fac.num_f
```

6.56 Modificar l'anterior per a aplicar també el descompte de la factura (amb l'àlies DESCOMPTE_2)

```
select linia_fac.num_f, data, cod_cli,  
SUM(quant*preu) as "IMPORT",  
SUM(quant*preu*(1-0.01*coalesce(linia_fac.dte,0))) AS "DESCOMPTE_1",  
SUM(quant*preu*(1-0.01*coalesce(linia_fac.dte,0))*(1-  
0.01*coalesce(factura.dte,0))) as "DESCOMPTE_2"  
from factura, linia_fac  
where factura.num_f = linia_fac.num_f  
group by linia_fac.num_f, data, cod_cli  
order by linia_fac.num_f
```

6.57 Traure el codi i nom d'aquells venedors que supervisen algú (consten com a cap). Traure també el número de supervisats de cadascun d'aquests supervisors.

```
select V1.cod_ven, V1.nom, count(V2.*)  
from Venedor V1, venedor V2  
where V1.cod_ven = V2.cod_cap  
group by V1.cod_ven, V1.nom
```

6.58 Traure el codi i descripció dels articles juntament amb el número de vegades que s'ha venut, el total d'unitats venudes i la mitjana d'unitats venudes per factura. Ordenar pel número total d'unitats venudes de forma descendent, i dins d'aquesta per codi d'article de forma ascendent.

```
SELECT LINIA_FAC.cod_a, ARTICLE.descrip, COUNT (*), SUM(quant), AVG(quant)  
FROM LINIA_FAC  
INNER JOIN ARTICLE USING(cod_a)  
GROUP BY LINIA_FAC.cod_a, ARTICLE.descrip  
  
ORDER BY 4 DESC, 1;
```

```
select article.cod_a, article.descrip, count(*),  
sum(linia_fac.quant), avg(quant)  
from article, linia_fac, factura  
where article.cod_a = linia_fac.cod_a  
and linia_fac.num_f = factura.num_f  
group by article.cod_a, article.descrip  
order by 4 desc, 1
```

6.59 Traure el codi i la descripció de les categories, amb la quantitat d'articles venuts de cada categoria, d'aquelles categories de les quals se n'han venut més de 100 unitats. Ordenar per aquest número de forma decendent.

```
select categoria.cod_cat, descripcio, sum(quant)  
from categoria, article, linia_fac  
where categoria.cod_cat = article.cod_cat  
and article.cod_a = linia_fac.cod_a  
group by categoria.cod_cat  
having sum(quant) > 100  
order by 3 desc
```

6.60 Traure el codi i el nom dels clients que no tenen cap factura.


```
select client.cod_cli, client.nom
from client left join factura on factura.cod_cli = client.cod_cli
where factura.cod_cli is null
```

6.61 Traure el codi, descripció i total d'unitats venudes de tots els articles, fins i tot d'aquells que no s'ha venut res.

Nota

Per a deixar-lo més bonic, com que la suma de valors nuls no és 0 sinó nul, per a que ens aparegue el valor 0 podem utilitzar la funció COALESCE(valor,0), que si el valor és nul torna un 0.

```
select article.cod_a, article.descrip, sum(coalesce(quant, 0))
from article left join linia_fac on linia_fac.cod_a = article.cod_a
group by article.cod_a
order by article.cod_a
```

6.62 Traure el nom de tots els pobles i el número de clients en cas de que en tinguin. Ordena per número de clients de forma descendent.

```
select poble.nom, count(client.cod_cli)
from poble left join client on poble.cod_pob = client.cod_pob
group by poble.cod_pob
order by 2 desc
```

6.63 Traure el codi i la descripció de les categories, amb el número d'articles de cada categoria i el número total d'unitats venudes de cada categoria, d'aquelles categories de les quals tenim més de 15 articles, i ordenat per número d'articles de forma descendent. Aquesta sentència ja és prou complicada. Concretament haureu de tenir en compte que:

- Voldrem traure el número d'articles de cada categoria, però potser alguns articles no s'han venut, i per tant no apareixeran en la taula LINIA_FAC.
- I també tenim el problema que, com ens fa falta la taula LINIA_FAC, un article venut en més d'una factura apareixerà més d'una vegada. Si comptem de forma normal, el comptaríem més d'una vegada cada article. Per tant voldrem comptar els diferents articles de cada categoria.

```
• select categoria.cod_cat, categoria.descripcio, count(distinct
  article.cod_a), sum(quant)
• from (article left join linia_fac on article.cod_a =
  linia_fac.cod_a ), categoria
• where categoria.cod_cat = article.cod_cat
• group by categoria.cod_cat
• having count(distinct article.cod_a) > 15
• order by count(article.cod_a) desc
```

6.64 Traure el número màxim de factures fetes a un client

```
select max(factures)
from (select count(num_f) as factures
      from factura
      group by cod_cli) as F;
```

6.65 Traure el l'import que suposa la factura més cara i l'import que suposa la més barata (sense considerar ni descomptes ni IVA)

```
select max(importeF), min(importeF)
from (select sum(quant*preu) as importeF
      from linia_fac lf
      group by num_f) as F;
```

6.66 Traure el número de factures més alt que s'ha venut per venedor en cada trimestre (no traurem qui és el venedor, que seria encara més complicat). Per a poder agrupar per trimestre, ens farà falta la funció **TO_CHAR(data,'Q')**, que trau el número de trimestre. El pas previ és calcular el número de factures de cada venedor i en cada trimestre. Després, amb la informació anterior, voldrem calcular el màxim de cada trimestre.

```
SELECT Trimestre, MAX(quantes)
FROM (SELECT TO_CHAR(data,'Q') AS Trimestre, cod_ven, count(*) AS quantes
FROM FACTURA
GROUP BY TO_CHAR(data,'Q'), cod_ven) AS F
GROUP BY Trimestre
ORDER BY 1;
```

```
select S.D, max(S.F)
from (select count(num_f) as F, cod_ven, to_char(data,'Q') as D
      from factura
      group by to_char(data,'Q'), cod_ven) as S
group by S.D
order by S.D
```

6.67 Traure els articles més cars que la mitjana. Tragueu-los ordenats per la categoria, i després per codi d'article.

```
select *
from article
where preu > (select AVG(preu)
              from article)
order by cod_cat, cod_a
```

6.68 Modificar l'anterior per a traure els articles més cars que la mitjana de la seua categoria. Tragueu-los ordenats per la categoria

```
select *
from article A1
where preu > (select avg(preu)
              from article
              where cod_cat = A1.cod_cat)
order by cod_cat, cod_a
```

6.69 Traure els pobles on tenim clients però no tenim venedors. Ha de ser per mig de subconsultes (en plural). Ordeneu per codi del poble.

```
SELECT *
FROM POBLE
WHERE NOT EXISTS (SELECT * FROM VENEDOR WHERE
POBLE.cod_pob=VENEDOR.cod_pob)
AND EXISTS (SELECT * FROM CLIENT WHERE
POBLE.cod_pob=CLIENT.cod_pob)
ORDER BY 1;
```

```
select *
from poble P1
where P1.cod_pob in (select P2.cod_pob
from poble P2 left join venedor on P2.cod_pob = venedor.cod_pob
where venedor.cod_ven is null)
and P1.cod_pob in (select P3.cod_pob
from poble P3, client
where P3.cod_pob = client.cod_pob)
order by P1.cod_pob
```

6.70 Traure els pobles on tenim més venedors que clients. Traure el codi del poble, el nom i el número de venedors. Ordena per nom del poble.

```
SELECT POBLE.cod_pob, POBLE.nom, count(*)
FROM POBLE INNER JOIN VENEDOR USING (cod_pob)
GROUP BY POBLE.cod_pob, POBLE.nom
HAVING count(*) > (SELECT count(*) FROM CLIENT WHERE
POBLE.cod_pob=CLIENT.cod_pob)
ORDER BY 2;
```

6.71 Traure l'import de la factura més cara de cada trimestre. La informació prèvia és la factura amb la data i l'import. A partir d'ahí haurem de calcular el màxim de l'import per a cada trimestre (no caldrà traure quina factura és).

```
SELECT Trimestre, MAX(import)
FROM (SELECT num_f, TO_CHAR(data,'Q') AS Trimestre, SUM(quant*preu) AS import
FROM LINIA_FAC INNER JOIN FACTURA USING (num_f)
GROUP BY num_f, Trimestre) AS LF
GROUP BY Trimestre
ORDER BY 1;
```

6.72 Traure el nom del venedor, el número de factures que ha venut i el percentatge que suposa sobre el total. Podria ser que en el moment de calcular el percentatge, el número resultant s'haja de convertir a numèric per a que dóna bé el resultat, ja que en fer una operació amb enters, el resultat serà enter. Aleshores hauríem d'obligar a que el número tinga decimals (::NUMERIC). I la funció d'arrodonir és **ROUND**. Ordeneu pel nom.

```

SELECT nom, COUNT(*),
ROUND(COUNT(*)/(SELECT count(*) FROM FACTURA)::NUMERIC*100,2)
FROM VENEDOR INNER JOIN FACTURA USING (cod_ven)
GROUP BY nom
ORDER BY 1;

```

6.73 Traure tota la informació (amb l'import) de la factura més cara. Ha de ser per mig de subconsultes. Mireu que segurament hi haurà 2 subconsultes. En la més interna calculem l'import de les factures. En l'altra calculem el màxim. I en la consulta principal, busquem la factura que coincideix amb aquest màxim.

```

SELECT num_f, data, import
FROM FACTURA
INNER JOIN (SELECT num_f, SUM(quant*preu) AS import
FROM LINIA_FAC GROUP BY num_f) AS LF1 USING(num_f)
WHERE import=(SELECT MAX(import)
FROM (SELECT num_f, SUM(quant*preu) AS import
FROM LINIA_FAC GROUP BY num_f) AS LF2);

```

6.74 (voluntari) Obtenir el venedor que ha venut més unitats de cada categoria, sense considerar en la categoria el valor nul. Aquesta consulta la podríem considerar ja com molt avançada.

```

SELECT cod_cat,VENEDOR.cod_ven, nom,SUM(quant) AS "Unitats Venudes"
FROM VENEDOR,FACTURA,LINIA_FAC,ARTICLE A
WHERE VENEDOR.cod_ven=FACTURA.cod_ven AND
FACTURA.num_f=LINIA_FAC.num_f
AND LINIA_FAC.cod_a=A.cod_a AND cod_cat IS NOT NULL
GROUP BY cod_cat,VENEDOR.cod_ven, nom
HAVING SUM(quant) >= ALL (SELECT SUM(quant)
FROM FACTURA,LINIA_FAC,ARTICLE
WHERE FACTURA.num_f=LINIA_FAC.num_f
AND LINIA_FAC.cod_a=ARTICLE.cod_a
AND cod_cat=A.cod_cat
GROUP BY cod_ven)
ORDER BY cod_cat;

```

6.75 Traure el nom de tots els clients i venedors implicats en alguna venda del primer trimestre de 2015. Intentar traure en una segona columna el text **Venedor** per als venedors, i **Client** per als clients. Ordenat pel nom.

```

select client.nom, 'Client'
  from client inner join factura using (cod_cli)
 where data BETWEEN '2015-01-01' and '2015-03-31'
union
select venedor.nom, 'Venedor'
  from venedor inner join factura using (cod ven)
 where data BETWEEN '2015-01-01' and '2015-03-31'
order by nom

```

6.76a Traure per mig de sentències d'operacions de conjunts els pobles on tenim algun venedor o algun client. No volem resultats repetits, i ho volem ordenat pel nom del poble.

```
select P1.nom
      from poble P1 inner join venedor using (cod_pob)
union
select P2.nom
      from poble P2 inner join client using (cod_pob)
order by nom
```

6.76b Modificar l'anterior per a traure els pobles on tenim al mateix temps venedors i clients

```
select P1.nom
      from poble P1 inner join venedor using (cod_pob)
intersect
select P2.nom
      from poble P2 inner join client using (cod_pob)
order by nom
```

6.76c Modificar l'anterior per a traure els pobles on tenim venedors però no tenim clients

```
select P1.nom
      from poble P1 inner join venedor using (cod_pob)
except
select P2.nom
      from poble P2 inner join client using (cod_pob)
order by nom
```

CONSULTAS DDL

6.77 Creeu la taula **CATEGORIA**, amb els mateixos camps i del mateix tipus que en la taula **CATEGORIA** de **FACTURA**, però de moment sense clau principal ni cap altra restricció. Guardeu la consulta de creació com **Ex_6_77.sql**

```
create table CATEGORIA
(cod_cat VARCHAR(15),
 descripcio VARCHAR(50))
```

6.78 Creeu la taula **ARTICLE**, també sense restriccions. Guardar la consulta com **Ex_6_78.sql**

```
create table ARTICLE
(cod_a VARCHAR(10),
 descrip VARCHAR(50),
 preu numeric(6, 2),
 stock numeric(4),
 stock_min numeric(4),
 cod_cat varchar(15))
```

6.79 Crear la taula **PROVINCIA**, amb la clau principal.

```
create table PROVINCIA
(cod_pro numeric(2) constraint cp_prov primary key,
nom VARCHAR(50))
```

6.80 Crear la taula **POBLE**, amb la clau principal i la restricció que el camp **cod_pro** és clau externa que apunta a PROVINCIA.

```
create table POBLE
(cod_pob numeric(5) constraint cp_poble primary key,
nom varchar(50),
cod_pro numeric(2) constraint ce_poble_prov references PROVINCIA)
```

6.81 Crear la taula **VENEDOR**, amb la clau principal i la clau externa a POBLE (de moment no definim la clau externa a VENEDOR, que és reflexiva).

```
create table VENEDOR
(cod_ven numeric(5) constraint cp_ven primary key,
nom varchar(100),
adreca varchar(100),
cp numeric(5),
cod_pob numeric(5) constraint ce_ven_poble references POBLE,
cod_cap numeric(5))
```

6.82 Crear la taula **CLIENT**, amb la clau principal i la clau externa a POBLE

```
create table CLIENT
(cod_cli numeric(5) constraint cp_cli primary key,
nom varchar(100),
adreca varchar(100),
cp numeric(5),
cod_pob numeric(5) constraint ce_cli_poble references POBLE)
```

6.83 Crear la taula **FACTURA**, amb la clau principal i les claus externes a CLIENT i VENEDOR. També heu d'exigir que **cod_cli** siga no nul.

```
create table FACTURA
(num_f numeric(5) constraint cp_fac primary key,
data date,
cod_cli numeric(5) constraint ce_fac_cli references client constraint
nn_cli not null,
cod_ven numeric constraint ce_fac_ven references venedor,
iva numeric,
dte numeric)
```

6.84 Crear la taula **LINIA_FAC**, amb la clau principal (observa que està formada per 2 camps) però de moment sense la clau externa que apunta a FACTURA ni la que apunta a ARTICLE. A més **cod_a** ha de ser no nul.

```
create table linia_fac
(num_f numeric(5),
num_l numeric(3),
cod_a varchar(10) constraint nn_lin_fac not null,
quant numeric(5),
preu numeric(6,2),
dte numeric(2),
constraint cp_lin_fac2 primary key (num_f, num_l))
```

6.85 Afegir un camp a la taula **VENEDOR** anomenat **alies** de tipus text, que ha de ser no nul i únic.

```
alter table venedor
add column alies text constraint nn_alies not null constraint u_alies
UNIQUE
```

6.86 Esborrar el camp anterior, **alies**, de la taula **VENEDOR**.

```
alter table venedor
drop column alies
```

6.87 Afegir la clau principal de **CATEGORIA**.

```
alter table categoria
add constraint cp_cat primary key (cod_cat)
```

6.88 En la taula **ARTICLE** afegir la clau principal i la clau externa a **CATEGORIA**.

```
alter table article
add constraint cp_article primary key (cod_a),
add constraint ce_art_cat foreign key (cod_cat) references categoria
(cod_cat)
```

6.89 En la taula **LINIA_FAC** afegir la clau externa que apunta a **FACTURA**, **exigint que s'esborre en cascada** (si s'esborra una factura, s'esborraran automàticament les seues línies de factura). I també la clau externa que apunta a **ARTICLE** (aquesta normal, és a dir NO ACTION)

```
alter table linia_fac
add constraint ce_lin_fac_fac foreign key (num_f) references factura
(num_f) on delete CASCADE,
add constraint ce_lin_fac_art foreign key (cod_a) references article
(cod_a)
```

6.90 Afegir un índex anomenat **i_nom_cli** a la taula **CLIENT** pel camp **nom**.

```
create index i_nom_cli on client (nom)
```

6.91 Afegir un índex anomenat **i_adr_ven** a la taula **VENEDOR** per a que estiga ordenat per **cp** (ascendent) i **adreca** (descendent).

```
create index i_adr_ven on venedor (cp, adreca desc nulls LAST)
```

6.92 Crear la vista **RESUM_FACTURA**, que ens dóni el total dels diners de la factura, el total després del descompte d'articles, i el total després del descompte de la factura, tal i com teníem en la consulta **6.56**. A partir d'aquest moment podrem utilitzar la vista per a traure aquestos resultats

```
create or replace view RESUM_FACTURA as
select linia_fac.num_f, data, cod_cli,
SUM(quant*preu) as "IMPORT",
SUM(quant*preu*(1-0.01*coalesce(linia_fac.dte,0))) as
"DESCOMPTE_1",
```

```
SUM(quant*preu*(1-0.01*coalesce(linia_fac.dte,0))*(1-
0.01*coalesce(factura.dte,0))) as "DESCOMPTE_2"
from factura, linia_fac
where factura.num_f = linia_fac.num_f
group by linia_fac.num_f, data, cod_cli
order by linia_fac.num_f
```

6.93 Inserir en la taula **CATEGORIA** les següents files:

cod_cat	descripcio
BjcOlimpia	Components Bjc Seria Olimpia
Legrand	Components marca Legrand
IntMagn	Interruptor Magnetotérmico
Niessen	Components Niesen Serie Lisa

```
insert into categoria
values ('BjcOlimpia', 'Components Bjc Seria Olimpia'),
      ('Legrand', 'Components marca Legrand'),
      ('IntMagn', 'Interruptor Magnetotérmico'),
      ('Niessen', 'Components Niesen Serie Lisa')
```

6.94 Inserir els següents articles.

cod_art	descrip	preu	stock	stock_min	cod_cat
B10028B	Cruzamiento Bjc Serie Olimpia	4.38	2	1	BjcOlimpia
B10200B	Cruzamiento Bjc Olimpia Con Visor	0.88	29		BjcOlimpia
L16550	Cartucho Fusible Legrand T2 250 A	5.89	1	1	Legrand
L16555	Cartucho Fusible Legrand T2 315 A	5.89	3	3	Legrand
IM2P10L	Interruptor Magnetotermico 2p, 4	14.84	2	1	IntMagn
N8008BA	Base Tt Lateral Niessen Trazo Bla	4.38	6	6	Niessen

```
insert into article
values ('B10028B', 'Cruzamiento Bjc Serie Olimpia', 4.38, 2, 1,
'BjcOlimpia'),
      ('B10200B', 'Cruzamiento Bjc Olimpia Con Visor', 0.88, 29, null,
'BjcOlimpia'),
      ('L16550', 'Cartucho Fusible Legrand T2 250 A', 5.89, 1, 1,
'Legrand'),
      ('L16555', 'Cartucho Fusible Legrand T2 315 A', 5.89, 3, 3,
'Legrand'),
      ('IM2P10L', 'Interruptor Magnetotermico 2p, 4', 14.84, 2, 1,
'IntMagn'),
```



```
('N8008BA', 'Base Tt Lateral Niessen Trazo Bla', 4.38, 6, 6, 'Niessen')
```

6.95 Inserir en la taula **CLIENT** tres files amb les següents dades

cod_cli	nom	adreca	cp	cod_pob
303	MIRAVET SALA, MARIA MERCEDES	URBANIZACION EL BALCO, 84-11		
306	SAMPEDRO SIMO, MARIA MERCEDES	FINELLO, 161	12217	
387	TUR MARTIN, MANUEL FRANCISCO	CALLE PEDRO VIRUELA, 108-8	12008	

```
insert into client (cod_cli, nom, adreca, cp)
values (303, 'MIRAVET SALA, MARIA MERCEDES', 'URBANIZACION EL BALCO, 84-11', null),
(306, 'SAMPEDRO SIMO, MARIA MERCEDES', 'FINELLO, 161', 12217),
(387, 'TUR MARTIN, MANUEL FRANCISCO', 'CALLE PEDRO VIRUELA, 108-8', 12008)
```

6.96 Inserir la següent factura:

num_f	data	cod_cli	cod_ven	iva	dte
6535	2015-01-01	306		21	10

num_f	num_l	cod_art	quant	preu	dte
6535	1	L16555	2	5.89	25

```
insert into factura
values (6535, '2015-01-01', 306, null, 21, 10)
```

```
insert into linia_fac
values (6535, 1, 'L16555', 2, 5.89, 25)
```

6.97 Inserir la següent factura (aquesta té més d'una línia de factura).

num_f	data	cod_cli	cod_ven	iva	dte
6559	2015-02-16	387		10	10

num_f	num_l	cod_art	quant	preu	dte
-------	-------	---------	-------	------	-----

6559	1	IM2P10L	3	14.84	
6559	2	N8008BA	6	4.38	20

```
insert into factura
values (6559, '2015-02-16', 387, null, 10, 10)
```

```
insert into linia_fac
values (6559, 1, 'IM2P10L', 3, 14.84, null),
(6559, 2, 'N8008BA', 6, 4.38, 20)
```

6.98 Esborrar la factura **6559**. Comprovar que també s'han esborrat les seues línies de factura

```
delete from factura
where num_f = 6559
```

6.99 Esborrar els articles dels quals **no** tenim **stock mínim**.

```
delete from article
where stock_min is null
```

6.100 Llevar tots els codis postals dels clients.

```
update client
set cp = null
```

6.101 Pujar el preu dels articles de la categoria **BjcOlimpia** un **5%** (el resultat serà que l'únic article d'aquesta categoria haurà passat d'un preu de 4.38 a **4.60€**)

```
update article
set preu = preu * 1.05
where cod_cat = 'BjcOlimpia'
```